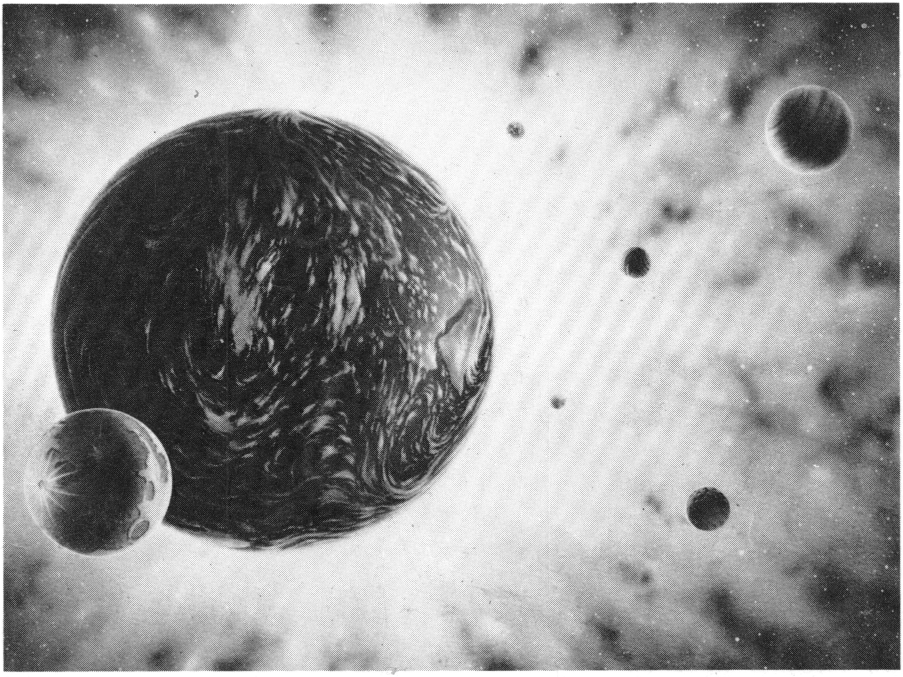
dB-SOFT

USED FOR
PC-8801mkII & PC-8801



COMPILER BOOK

(dB-IBASIC用)

8801

dB-IBASIC & dB-COMPILER SERIES

はじめに

dB－ＳＯＦＴは、パーソナルコンピュータが誰にでも広く活用されることを願い、その利用技術の開発に力を注いでいるソフトウェアのトータルクリエーションハウスです。

このコンパイラブックは、当社が〝dB－BASIC〟の基本思想にもとづき、コンパチビリティーを大切に開発してきた「dB－コンパイラ」の操作説明書です。これは、かねてより使い易さの一方で実行速度の遅さの点で不満の多かった、インタプリタに対する回答として、当社の技術陣が開発したプログラムです。このパッケージのもうひとつのシステムプログラム「dB－ＩＢＡＳＩＣ」と併用することにより、大きな威力を発揮します。ですから、今まで操作性の複雑さから敬遠されていたマシン語プログラムが簡単に取り組めるようになりました。スピード感を要するゲームやパズルなども短期間で作成できます。

ＰＣ－8801の機能を知りつくしたdB－ＳＯＦＴの「dB－ＩＢＡＳＩＣ」と「dB－コンパイラ」によって、あなたのオリジナルプログラミングワールドを大きく広げてください。パソコンが更に楽しくなることでしょう。併せて、他ハードのdB－IBASIC版プログラムのコンバージョンも、ぜひお試しください。

昭和59年２月１日

古谷　貞行

ご利用の皆様へ

「dB－IBASIC」と「dB－コンパイラ」は、dB－SOFTが自らの開発ツールとして、十分にその機能を検証したシステムプログラムです。そのため、両システムともプロフェッショナル・ニーズに応える種々の特長があります。

「dB－コンパイラ」は、プログラムの最適化によって実行速度や効率が格段に高いものになっています。また、メモリを多くとりがちな制約を緩和するために、G-VRAMにオブジェクトを落とせるので、オブジェクトを約30Kバイトもの広さまでコンパイルできます。プログラミングやデバックの容易なIBASICと、高速処理化するコンパイラを使って、スピード感あふれるゲームプログラムを多数作られることを期待しています。

ご注意

「dB－コンパイラ」でコンパイルしたソフトを、ランタイムルーチン付きの商品として販売する場合、当社の著作権に低触することになります。もし、商品化される場合は、あらかじめ「dB－コンパイラ許諾契約」等の資料をご請求ください。

資料請求先　〒060　札幌市中央区北３条西４丁目カミヤマビル
デービーソフト「dB－コンパイラ」係

コンパイルの手順

ソースプログラムは、dB-IBASICのインタプリタ上で十分にBUG（バグ）を取ってください。

1．コンパイラ本体はマシン語で組み込まれていますので、モニタ（MON）モードで次のようにロードしてください。

```
h]R2:⏎
```

2．ロード終了後、〔h]〕GC000 ⏎ と入力します。（〔　〕内はあらかじめ表示されます）

なお、次のテープロードで1200ボーでセーブしたソーステープをロードする場合、実行する前にh]S9806 ⏎ として9806Hに1をセットしてください。また、600ボーに戻す場合は、9806Hに0をセットしてください。

3．入力を終えるとCRT画面に、次のように表示されます。

```
*I-BASIC COMPILER Ver 1.0*
 TAPE LOAD (Y or N) ?
```

カセットテープからソースプログラムをロードするか、メインメモリのテキスト上にあるソースプログラムをそのまま使用するかを決めてください。テープからソースプログラムをロードする場合、カセットテープをセットしてからYを入力してください。

注意☞　すでにメインメモリ上にソースプログラムがあり、それをコンパイルする場合は、テープからロードする必要がありませんのでNを入力してください。

☞　コンパイラをロードした場合、モニタからCTRL＋Bでインタプリタ等に復帰することはできません。

4．次に、テープからロードしたソースプログラム、またはすでにメインメモリ上にあるソースプログラムのテキストの範囲が、16進数の形式で表示されます。TEXT　STARTに続く値が、テキストの先頭番地となります。TEXT　ENDに続く値が、テキストの最終番地となります。

5．続いて、ソースプログラムをコンパイルしてオブジェクトに落とす番地（ADR）を聞いてきます。

OBJ　ADR（<8000H）？　$\boxed{0}$000

↑ **カーソル点滅**

ここでオブジェクトの先頭番地を設定しますが、設定可能番地は0～7FFFHです。

注☞　**3**．で述べたようにモニタモードよりdB-IBASICのインタプリタレベルに戻ってソースプログラムの再編集は不可能です。

6．オブジェクトに落とす先頭アドレスを入力すると、CRT画面に次のように表示されます。

PORT　SAVE（Y or N）？

コンパイルによってオブジェクトに落とす場所を指定します。$\boxed{Y}$を入力するとG-VRAM上にコンパイルしたオブジェクトプログラムを落とします。（I／Oポートを経由してG-VRAM上にセーブする）$\boxed{N}$を入力すると**5**．で設定したメインメモリ上のオブジェクトの先頭番地より、オブジェクトプログラムがセーブされます。

ソースプログラムのテキスト領域と、オブジェクトの領域が重なっていない場合のみ、直接メインメモリにオブジェクトを落とせます。

ソースプログラムのテキスト領域にオブジェクトが重なっている場合は、G-VRAMにセーブ（PORT　SAVE）します。

7．続いて、次のようにコンパイラのワーク先頭番地を聞いてきます。

```
WORK  ADR.（<9800H）？  [0]000
                         ↑
                         └── カーソル点滅
```

9800$_{\mathrm{H}}$より小さい値です。ソースプログラムのテキスト領域や、メインメモリにオブジェクトを落とす場合、さらにそのオブジェクト範囲に重ならない番地を指定します。また、疑似PCGを使う場合、9000$_{\mathrm{H}}$以上にします。（PCGデータエリア7800$_{\mathrm{H}}$～8FFF$_{\mathrm{H}}$）ソースプログラム（テキスト）の大きさによってオブジェクトの大きさも変わりますが、通常は9000$_{\mathrm{H}}$～9500$_{\mathrm{H}}$くらいがよいでしょう。

8．さらに次のように聞いてきます。

```
LINE  TABLE  DISPLAY （Y or N）？
```

ソースプログラムの行番号に対応するオブジェクトプログラムの番地を、画面に表示する場合は$\boxed{Y}$を、表示しない場合は$\boxed{N}$を入力してください。
（通常は$\boxed{N}$を入力してください）

続けて、$\boxed{⏎}$キーを押すとコンパイルを開始し、次のように表示します。

```
PASS-1
PASS-2
```

（番地の表示を指定した場合、行番号とそれに対応する番地も表示されます）

この時エラーが出た場合、ソースプログラムにコンパイルできない箇所があります。dB-IBASICを再びロードし、再度ソースプログラムのリストを見直してください。
（ex. $L, LIST, CLEAR, etcこれらはコンパイルできません）

9．コンパイルが終了すると、オブジェクトプログラムの最終番地と、変数の最終番地が次のように表示されます。

OBJ END ADR.＝□□□□H ………オブジェクト
VAR END ADR.＝□□□□H ………変数

10．オブジェクトをG-VRAMに落とした場合（**6**．のPORT SAVEでYを入力した場合）、さらに次のように聞いてきます。

PORT LOAD（Y or N）？

G-VRAMに落としたオブジェクトプログラムをメインメモリにロードする場合はYを、ロードしない場合はNを入力してください。Nを入力すると、メインメモリ内のソースプログラム等は、破壊されずそのまま残ります。

11．コンパイルが終了し各最終番地が表示されると（あるいはPORT LOAD（Y or N）？ を入力後）、モニタのコマンド待ちモードとなり、プロンプトシンボル**h]**が表示されます。それぞれのモニタのコマンドで必要な操作を行ってください。

コンパイルしたオブジェクトプログラムを実行する場合は、必ず〔**h]**〕**G 9803**と入力します。

dB－IBASICの命令、およびコンパイル・チェックポイント

◆コマンド

APPEND

メモリに格納されている行番号の最終値より、テープから追加する行番号の方が大きくなければなりません。

RUN

RUN命令をコンパイルしたオブジェクトプログラムでのRUN実行は、先頭アドレス、または指定参照先に対応するアドレスに飛ぶだけです。変数の初期化等はインタプリタレベルと違い、行いません。変数の初期化が必要な時は、必ずCLR命令を使ってください。

◆特殊コマンド

$L

初期値は15文字です。したがって文字列としては1変数に対して14文字までしか扱えません。それ以上使用する場合は、IBASICが起動した直後（コンパイルする場合もIBASICの起動時）に必ず$Lを実行し、使用する文字列の文字数（最大255まで）を設定してください。

◆一般ステートメント

CASE

条件にELSEは使えません。

CLEAR

dB-IBASICは$Lで各文字列の長さを固定しています。CLEARはdB-IBASICのインタプリタレベルのみで有効です。コンパイルはできません。変数の初期化のみ行う場合、CLRを使います。

DIM

2次元配列まで、添字は0～254の範囲です。

END

この命令をコンパイルし実行すると、ENDに対応する場所で実行を停止してモニタモードとなります。

FOR～NEXT

dB-IBASICではSTEPの増分において、変数指定できません。必ず数値定数を使います。

GOTO/GOSUB/RESTOR等

GO␣TOは不可。ラベルは（〝）でくくって指定します。
(ex. GOTO 〝DATA PUT〟)

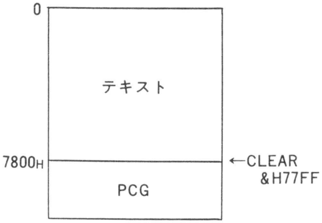
STOP

コンパイルしてこの命令が実行された時、STOP が押されていればENDと同じ動作を行います。それ以外は次の実行に移ります。

DEFCHR$

DEFCHR$で、疑似PCGのデータを定義する場合、CLEAR文でテキストエリアの上限を$77FF_H$に設定してPCGのデータエリアを必ず確保してください。(ただし、インタプリタレベルのみです)

※確保しないと、テキストとPCGデータが重なったり、スタックが壊れたりして誤動作を起こします。



◆入出力ステートメント

INPUT

複数の変数指定はできません。

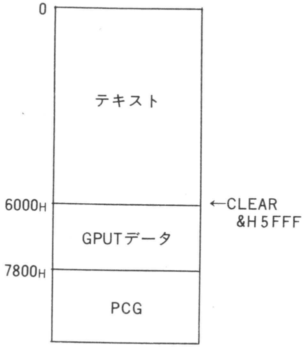
PRINT

dB-IBASICではCRT画面に対するPRINT # 0のみです。

GPUT/MPUT

GPUT/MPUTするデータは、メモリの空エリアにMEMやPOKE文で書き込むわけですが、データエリアの確保が必要です。GPUTで疑似PCGをPUTする場合、空エリアは7800_Hより前のアドレスとなります。たとえば6000_HからGPUTするデータを入れたとすれば、CLEAR文でテキストの上限を$5FFF_H$に設定し、6000_H以降の各データエリアを必ず確保します。

※疑似PCG使用や、MPUT、GPUTでのデータエリア確保の為、CLEAR文でテキストの上限が限定されますから当然テキストのフリーエリアは少なくなります。



◆スクリーンステートメント

COLOR

dB-IBASICでは、すべてのパラメータを省略するとコンパイラでは機能しません。

LINE

dB-IBASICでは、タイルパターンが使えます。

PAINT

必ず境界色を指定しなければなりません。（コンパイラの場合）

◆特殊ステートメント

USR

dB-IBASICでは、関数ではなくステートメントです。USRの後に直接、値をつけて実行させます。

WAIT

引数1に対して1msec停止する。

◆一般関数

FRE

dB-IBASICでは、式はダミーでテキストの空エリアを与えます。

POS

dB-IBASICではX、Y両方の座標を与えることができます。N88-BASICのCSRLINの代用をPOS（1）で行います。

数値関数

小数点以下がでるものは、小数点以下を切り捨て、整数部のみを与えます。

SIN/COS

dB-IBASICでは度単位の値を与え、また何倍するかも指定します。

RND

dB-IBASICでは0から（式の値）－1までの値の整数を与えます。

その他

FOR～NEXT/GOSUB etc スタックの多用

実行回数がインタプリタレベルよりコンパイルオブジェクトの方がはるかに多いため、LOGICAL ERROR 等が生じやすいので数値があまりたまらないようにします。

行番号を参照する命令（GOTO/GOSUB etc）がコンパイルできない事があります。指定の行番号に対応するアドレスと、実際のアドレスがずれたまま補正されていない場合に起こります。コンパイルする前にインタプリタ上でソースを編集（ ↵ キーを押して登録）する事により、IBASICインタプリタが補正します。これをソースとしてコンパイルします。

ミニフラッピーのロードとコンパイル

dB-IBASICテープのB面には、ミニフラッピーのソースプログラムが入っています。プログラムは、疑似PCGに対するキャラクタデータの定義部分と、ゲームのメインプログラムの2つが続いて入っています。

① IBASICを起動させます。（リファレンスブック参照）

② 起動後、まずPCGデータのプログラムをロードします。

```
LOAD ⏎
```

③ ロード終了後、PCGエリアをCLEAR文で確保します。

```
CLEAR &H5FFF ⏎
```

※PCGのデータエリアの確保だけなら、アドレスは77FFHですが、ここではGPUTのデータエリアを6000Hより使用しているので5FFFHをテキストの上限としています。

④ プログラムを実行し、疑似PCGデータメモリ（7800H～8FFFH）に定義します。

```
RUN ⏎
```

⑤ しばらくしてOkが出て定義が終了すると、このPCGのデータプログラムは不要になるので消します。

```
NEW ⏎
```

⑥ 次に、ゲームのメインプログラムをロードします。

```
LOAD ⏎
```

ロード終了後、RUNするとdB-IBASICのインタプリタレベルでゲームは動きますが、このインタプリタのゲームは内部処理が複雑ですと非常に動きが遅く、ゲームのおもしろみに欠ける場合があります。そこで行うのが**「コンパイル」**です。

ミニフラッピーをコンパイルしましょう。

※ロード終了後、RUNした場合は STOP キーでプログラムを止め、CLS4 ↵ で画面を消してください。

① まずモニタモードにします。"h]"が頭に表示され、モニタモードになります。

MON ↵

② dB-コンパイラテープをセットし、ロードします。 h]R2: ↵

③ ロード終了後、C000H番地に飛ばしてコンパイラを実行させます。

h]GC000 ↵

④ ＊ I-BASIC COMPILER Ver 1.0 ＊

TAPE LOAD (Y or N) ?

すでにIBASICのインタプリタ上でメインプログラムをロードしているのでメモリにソースがありますからNを入力します。

⑤ OBJ ADR. (<8000H) ?0000

次にコンパイルしたオブジェクトの先頭アドレスは、ミニフラッピーソースが大きいので同様にオブジェクトも大きくなります。したがって0番地からでなければ入りませんので0000のまま ↵ を押します。

⑥ PORT SAVE (Y or N) ?

テキストのプログラム（ソース）とオブジェクトのアドレスが重なっていますので、当然G-VRAM上にコンパイルしたオブジェクトをポートセーブしなければなりませんのでYを入力します。

⑦ WORK ADR. (<9800H) ?0000

9000 ↵ としてください。9800Hより小さいアドレスが必要なのですが、0～8FFFHまではそれぞれソーステキスト、GPUTのデータ、PCGのデータが入っていて使えません。必ず9000Hにします。

⑧ LINE TABLE DISPLAY (Y or N) ?

続いて、ソースの行番号に対応するアドレスを表示するかどうかを聞いてきます。通常Nを入力してください。

これでコンパイルを開始し、オブジェクトはG-VRAM上にセーブされます。

（G-VRAMに書き込まれるので画面にグラフィックのドットが表示されます）

⑨ PASS-1

PASS-2

オブジェクト、および変数のエンドアドレスが表示されます。

PORT LOAD (Y or N) ?

🄽を入力してください。0番地からオブジェクトがロードされます。

⑩ 再びモニタモードになります。(h]が表示されます) h]G9803 ⏎

コンパイルされたミニフラッピーのオブジェクトプログラムが実行されます。

テープロードによるコンパイル

まずコンパイラを起動させます。

① TAPE LOAD (Y or N) ?

🅈を入力し、dB-IBASICテープB面のPCGのデータをロードします。

② OBJ ADR. (<8000H) ?0000 ⏎

③ PORT SAVE (Y or N) ?Y ⏎

④ WORK ADR. (<9800H) ?9000 ⏎

⑤ LINE TABLE DISPLAY (Y or N) ?N ⏎

⑥ コンパイル終了後、PORT LOAD (Y or N) ?Y ⏎

⑦ モニタモードに戻ったら、オブジェクトを実行します。h]G9803 ⏎

⑧ 実行後、再びモニタモードなります。再びコンパイラをロードし実行します。

h]R2: ⏎

h]GC000 ⏎

⑨ TAPE ROAD (Y or N) ?Y ⏎

dB-IBASIC B面のミニフラッピーのメインプログラムをロードします。

⑩ OBJ ADR. (<8000H) ?0000 ⏎

⑪ PORT SAVE (Y or N) ?Y ⏎

⑫ WORK ADR. (<9800H) ?9000 ⏎

⑬ LINE TABLE DISPLAY (Y or N) ?N ⏎

⑭ コンパイル終了後、PORT LOAD (Y or N) ?Y ⏎

⑮ モニタモードで、h]G9803 ⏎

ゲームが開始されます。

※dB-コンパイラテープのB面には、コンパイラ版ミニフラッピーが入っています。ロードの方法は次の通りです。

```
]R2:↵
]GC000 ↵
```

再び、マシン語がロードされ、ロード終了後ゲームが開始されます。

これはコンパイラをロードせずに実行できます。ただし、dB-コンパイラの著作上の理由から公開しておりませんのでご了承ください。

通常はソースプログラムをdB-コンパイラでコンパイルしてください。

キャラクタデータリスト

```
1 DEFCHR$(&H00)=HEXCHR$("0000000000000000000E1F1F3F3F3F1F000E1F1F3F3F3F1F")
2 DEFCHR$(&H01)=HEXCHR$("000000000000000000000F3F3F3F1F1F00000F3F3F3F1F1F")
3 DEFCHR$(&H02)=HEXCHR$("000000000000000000000000030F1F17000000000030F1F17")
4 DEFCHR$(&H03)=HEXCHR$("000000000000000037373F1F3F3F3F0037363F1F3F3F3F00")
5 DEFCHR$(&H04)=HEXCHR$("00000000000106300000000000000000006193F000000030")
6 DEFCHR$(&H05)=HEXCHR$("78003060F8F410187F077F1E060B0C2078003060F8F47018")
7 DEFCHR$(&H06)=HEXCHR$("000000000000000100000000000007 1E0000000000000000")
8 DEFCHR$(&H07)=HEXCHR$("0630780070FCF01C397F7F071F020F000030780070FCF07C")
9 DEFCHR$(&H08)=HEXCHR$("00000101011A0E0408142A048A54A05100000101011A0E04")
10 DEFCHR$(&H09)=HEXCHR$("0C1C001030010100A201A8448A142A150C1C001030010100")
11 DEFCHR$(&H0A)=HEXCHR$("0006334F4F362301000030484838142A0006334F4F362301")
12 DEFCHR$(&H0B)=HEXCHR$("213103020B720004944AA450A000AA55213103020B720000")
13 DEFCHR$(&H0C)=HEXCHR$("0006334F4F3E3F7F00003048483000000000006334F4F3E3F67")
14 DEFCHR$(&H0D)=HEXCHR$("7FFEC60008381018000128 58A247AA451F1E060008381018")
15 DEFCHR$(&H0E)=HEXCHR$("0006334F4F3E3F7F00003048483000000000006334F4F3E3F67")
16 DEFCHR$(&H0F)=HEXCHR$("7FFFFC000C7E7E3C00000000000000001F1F04600C7E7E3C")
17 DEFCHR$(&H10)=HEXCHR$("00000000000000000000F0FCFCFCF8F80000F0FCFCFCF8F8")
18 DEFCHR$(&H11)=HEXCHR$("00000000000000000070F8F8FCFCFCF80070F8F8FCFCFCF8")
19 DEFCHR$(&H12)=HEXCHR$("000000000000000000000000C0F0F8E800000000C0F0F8E8")
20 DEFCHR$(&H13)=HEXCHR$("0000000000000000ECECFCF8FCFCFC00EC6CFCF8FCFCFC00")
21 DEFCHR$(&H14)=HEXCHR$("00000000000000600C00000000E098FC000000000000000C")
22 DEFCHR$(&H15)=HEXCHR$("1E000C061F2F0818FEE0FE7860D030041E000C061F2F0E18")
23 DEFCHR$(&H16)=HEXCHR$("000000000000001000000000000E0E8000000000000000000")
24 DEFCHR$(&H17)=HEXCHR$("600C1E000E3F0F389CFEFEE0F840F000000C1E000E3F0F3E")
25 DEFCHR$(&H18)=HEXCHR$("00008080C0702000A854285122058A4900508890C2742208")
26 DEFCHR$(&H19)=HEXCHR$("001030A000800000A245CA152A542810021130A000800000")
27 DEFCHR$(&H1A)=HEXCHR$("0060CCF24204048400000C12AA54A8510060CCF242040484")
28 DEFCHR$(&H1B)=HEXCHR$("D0B00008182E00002A07A2572200AA55D0B00008182E0000")
29 DEFCHR$(&H1C)=HEXCHR$("0060CCF2F27CFCFE00000C12120C00000060CCF2F27CFCE6")
30 DEFCHR$(&H1D)=HEXCHR$("FE7F1B00107460220000E452A50A9E14F878000410746022")
31 DEFCHR$(&H1E)=HEXCHR$("0060CCF2F27CFCFE00000C12120C00000060CCF2F27CFCE6")
32 DEFCHR$(&H1F)=HEXCHR$("FEFF3F00307E7E3C00000000000000000F8F82006307E7E3C")
33 DEFCHR$(&H20)=HEXCHR$("000000000000000000000000000000000000000000000000")
34 DEFCHR$(&H21)=HEXCHR$("0000000000000000000000183C7E18300000000000001830")
35 DEFCHR$(&H22)=HEXCHR$("00FCFEDADADADA0000FCFEDADADADA0000FCFEDADADADA00")
36 DEFCHR$(&H23)=HEXCHR$("00EEEEFEFEEEEE0000EEEEFEFEEEEE0000EEEEFEFEEEEE00")
37 DEFCHR$(&H24)=HEXCHR$("007C383838387C00007C383838387C00007C383838387C00")
38 DEFCHR$(&H25)=HEXCHR$("000000383800000000000000383800000000000038380000000")
39 DEFCHR$(&H26)=STRING$(3,HEXCHR$("7CFEF87C3EFE7C00"))
40 DEFCHR$(&H27)=STRING$(3,HEXCHR$("007CE6E0E6FE7C00"))
41 DEFCHR$(&H28)=STRING$(3,HEXCHR$("007CE6E6E6FE7C00"))
42 DEFCHR$(&H29)=STRING$(3,HEXCHR$("00FCEEEEFCFEEE00"))
43 DEFCHR$(&H2A)=STRING$(3,HEXCHR$("007EF8FEF8FE7E00"))
44 DEFCHR$(&H2B)=STRING$(3,HEXCHR$("007EF8FEF8F87800"))
```

```
45 DEFCHR$(&H2C)=STRING$(3,HEXCHR$("00FCFEE6E6E6E600"))
46 DEFCHR$(&H2D)=STRING$(3,HEXCHR$("3C4299A1A199423C"))
47 DEFCHR$(&H2E)=STRING$(3,HEXCHR$("00063E7E66663E00"))
48 DEFCHR$(&H2F)=STRING$(3,HEXCHR$("0000E0F9E5E4F900"))
49 DEFCHR$(&H30)=STRING$(3,HEXCHR$("7CEEEEEEEEEE7C00"))
50 DEFCHR$(&H31)=STRING$(3,HEXCHR$("3878383838387C00"))
51 DEFCHR$(&H32)=STRING$(3,HEXCHR$("7CEE0E3C70FEFE00"))
52 DEFCHR$(&H33)=STRING$(3,HEXCHR$("7CEE0E3C0EEE7C00"))
53 DEFCHR$(&H34)=STRING$(3,HEXCHR$("3C7CDCDEFE1C1C00"))
54 DEFCHR$(&H35)=STRING$(3,HEXCHR$("FEE0FC0E0EEE7C00"))
55 DEFCHR$(&H36)=STRING$(3,HEXCHR$("7CEEE0FCEEEE7C00"))
56 DEFCHR$(&H37)=STRING$(3,HEXCHR$("FEEE0E1C38383800"))
57 DEFCHR$(&H38)=STRING$(3,HEXCHR$("7CEEEE7CEEEE7C00"))
58 DEFCHR$(&H39)=STRING$(3,HEXCHR$("7CEEEE7E0EEE7C00"))
59 DEFCHR$(&H3A)=STRING$(3,HEXCHR$("00FEFE3838383800"))
60 DEFCHR$(&H3B)=STRING$(3,HEXCHR$("007CFEE6E6FEE600"))
61 DEFCHR$(&HA8)=HEXCHR$("0000000000000000FEFEFE00EFEFEF000000000000000000")
62 DEFCHR$(&HA9)=HEXCHR$("FEFEFE00EFEFEF0000000000000000000000000000000000")
63 DEFCHR$(&H3C)=HEXCHR$("FCFE667C66FEFC00FCFE667C66FEFC00FCFE667C66FEFC00")
64 DEFCHR$(&H3D)=HEXCHR$("000000B0F676E000000000B0F676E000000000B0F676E000")
65 DEFCHR$(&HB8)=HEXCHR$("000000000000000000000000071F1737000000000071F1717")
66 DEFCHR$(&HB9)=HEXCHR$("0000000000000000371F1F1F1F1F3F1C371F1F1F1F1F3F1C")
67 DEFCHR$(&HBC)=HEXCHR$("0000000000000000000000071F173737000000071F171737")
68 DEFCHR$(&HBD)=HEXCHR$("00000000000000001F1F1F1F1F0F03071F1F1F1F1F0F0307")
69 DEFCHR$(&H44)=HEXCHR$("00003060400000000006337F7F3F3F7F0000306040000000")
70 DEFCHR$(&H45)=HEXCHR$("80808000000000007F7F7F3F03787E3C8080804000000000")
71 DEFCHR$(&H46)=HEXCHR$("003060400000000006337F7F3F3F7F7F0030604000000000")
72 DEFCHR$(&H47)=HEXCHR$("00406000000000007F3F03000C0000000040606000000000")
73 DEFCHR$(&H48)=HEXCHR$("003060400000808006337F7F3F3F7F7F0030604000008080")
74 DEFCHR$(&H49)=HEXCHR$("80000000000000007F3F031C3E3F3F1E8040000000000000")
75 DEFCHR$(&H4A)=HEXCHR$("071F3F7F7FFFFFFF071F3F7F7FFFFFFF071F3F7F7FFFFFFF")
76 DEFCHR$(&H4B)=HEXCHR$("FFFFFF7F7F3F1F07FFFFFF7F7F7F7F3FFFFFFF7F7F3F1F07")
77 DEFCHR$(&H4C)=HEXCHR$("1820377E6C3563DF1826377F6F3F7FE71820377E6C3563DF")
78 DEFCHR$(&H4D)=HEXCHR$("ECEEFF9A7F73391EFFBF379A7F7F7F3EECEEFFFA7F73391E")
79 DEFCHR$(&H4E)=HEXCHR$("28423CF89174231C28463FFF977E3F7728423CF89174231C")
80 DEFCHR$(&H4F)=HEXCHR$("E4F0FC28C4432418DF9F4C28CC7F7E3CE4F0FC68C4432418")
81 DEFCHR$(&HBA)=HEXCHR$("000000000000000000000000C0F0F8F800000000C0F0F8F8")
82 DEFCHR$(&HBB)=HEXCHR$("000000000000000098E8E898F8F8FC3898E8E898F8F8FC38")
83 DEFCHR$(&HBE)=HEXCHR$("0000000000000000000000C0F0F8F8F8000000C0F0F8F8F8")
84 DEFCHR$(&HBF)=HEXCHR$("0000000000000000F8D8E8E818F0C080F8D8E8E818F0C080")
85 DEFCHR$(&H54)=HEXCHR$("00000C26120000000060CCFEFEFCFCFE00000C2612000000")
86 DEFCHR$(&H55)=HEXCHR$("0101010000000000FEFEFEFCC01E7E3C0101010200000000")
87 DEFCHR$(&H56)=HEXCHR$("000C26120000010160CCFEFEFCFCFEFE000C261200000101")
88 DEFCHR$(&H57)=HEXCHR$("0100000000000000FEFCC0387CFCFC780102000000000000")
89 DEFCHR$(&H58)=HEXCHR$("000C26120000000060CCFEFEFCFCFEFE000C261200000000")
90 DEFCHR$(&H59)=HEXCHR$("0002060000000000FEFCC000300000000002060600000000")
91 DEFCHR$(&H5A)=HEXCHR$("E0F8FCFEFEFFFFFFE0F8FCFEFEFFFFFFE0F8FCFEFEFFFFFF")
92 DEFCHR$(&H5B)=HEXCHR$("FFFFFFFEFEFCF8E0FFFFFFFEFEFEFEFCFFFFFFFEFEFCF8E0")
93 DEFCHR$(&H5C)=HEXCHR$("70DC0EF67BBDCE7E70FCCEF6FBFDFEE670DC0EF67BBDCE7E")
94 DEFCHR$(&H5D)=HEXCHR$("167FFFE316E6DC38FCFDF9E336FEFE3C167FFFE716E6DC38")
95 DEFCHR$(&H5E)=HEXCHR$("58860D5FB21D885C58E6CDFFF27DFCEE58860D5FB21D885C")
96 DEFCHR$(&H5F)=HEXCHR$("06377FC3028C18B0FAFC65C332FE7EBC06377FC7028C18BC")
97 DEFCHR$(&H60)=HEXCHR$("20403048C83408982046334FCF3E3FE720403048C8340898")
98 DEFCHR$(&H61)=HEXCHR$("60E0F800800420001F1F04008C7E7E3C60E0F86080042000")
99 DEFCHR$(&H62)=HEXCHR$("00003049483000C000C07ECF4FFF7F3F00003049483000C0")
100 DEFCHR$(&H63)=HEXCHR$("3E3FFF0E000000001000000040F0FC783E3FFF3F0E000000")
101 DEFCHR$(&H64)=HEXCHR$("00304848300000FEC07ECF4FFF7F3F0100304848300000FE")
102 DEFCHR$(&H65)=HEXCHR$("FF3F0E00000000000000000001070707FF3F3F0E00000000")
103 DEFCHR$(&H66)=HEXCHR$("0000000003040000000000030F1F3F3F0000000003040000")
104 DEFCHR$(&H67)=HEXCHR$("00030100000000007F7CFC7800F8FC780003030100000000")
105 DEFCHR$(&H68)=HEXCHR$("000000020C1000000000030F1F3F3F7F000000020C100000")
106 DEFCHR$(&H69)=HEXCHR$("03010000000000007CFC7E00030707030303010000000000")
107 DEFCHR$(&H6A)=HEXCHR$("00003048483000180006334F4F3E3F670000304848300018")
108 DEFCHR$(&H6B)=HEXCHR$("60E0F800000000001F1F04000C7E7E3C60E0F86000000000")
109 DEFCHR$(&H6C)=HEXCHR$("003048483000000006334F4F3E3F7F7F0030484830000000")
110 DEFCHR$(&H6D)=HEXCHR$("78F8E000000000000400000000E0000078F8E060000000000")
111 DEFCHR$(&H6E)=HEXCHR$("00304848300078F806334F4F3E3F070700304848300078F8")
112 DEFCHR$(&H6F)=HEXCHR$("E0000000000000001C1800001E3F3F1EE060000000000000")
113 DEFCHR$(&H70)=HEXCHR$("10820C93520C019810E2CCF3F27CFDE610820C93520C0198")
114 DEFCHR$(&H71)=HEXCHR$("061F1F4000040021F9F0FD40307E7E39061F1F4600040021")
115 DEFCHR$(&H72)=HEXCHR$("00000000C0200000000000C0F0F8FC00000000C020000000")
```

```
116 DEFCHR$(&H73)=HEXCHR$("00C0800000000000FE3E3F1E001F3F1E00C0C08000000000")
117 DEFCHR$(&H74)=HEXCHR$("00000040300800000000C0F0F8FCFCFE0000004030080000")
118 DEFCHR$(&H75)=HEXCHR$("C0800000000000003E3F7E00C0E0E0C0C0C0800000000000")
119 DEFCHR$(&H76)=HEXCHR$("00000C92120C000300037EF3F2FFFEFC00000C92120C0003")
120 DEFCHR$(&H77)=HEXCHR$("7CFCFF700000000800000020F3F1E7CFCFFFC70000000000")
121 DEFCHR$(&H78)=HEXCHR$("000C12120C00007F037EF3F2FFFEFC80000C12120C00007F")
122 DEFCHR$(&H79)=HEXCHR$("FFFC70000000000000000000080E0E0FFFCFC70000000000")
123 DEFCHR$(&H7A)=HEXCHR$("00000C12120C00180060CCF2F27CFCE600000C12120C0018")
124 DEFCHR$(&H7B)=HEXCHR$("06071F0000000000F8F82000307E7E3C06071F0600000000")
125 DEFCHR$(&H7C)=HEXCHR$("000C12120C001E1F60CCF2F27CFCE0E0000C12120C001E1F")
126 DEFCHR$(&H7D)=HEXCHR$("07000000000000003818000078FCFC780706000000000000")
127 DEFCHR$(&H7E)=HEXCHR$("000C12120C00000060CCF2F27CFCFEFE000C12120C000000")
128 DEFCHR$(&H7F)=HEXCHR$("1E1F07000000000020000000700000001E1F070600000000")
129 DEFCHR$(&H80)=HEXCHR$("3048C8F0D8E0E078364BCFFEE71F1F043048C8F0D8E0E078")
130 DEFCHR$(&H81)=HEXCHR$("3048C8F0C00000003148C8F0C0000000314FCFFEFF3F3F1F")
131 DEFCHR$(&H82)=HEXCHR$("000000000000304000000002030003040000000203070F1C")
132 DEFCHR$(&H83)=HEXCHR$("040300000000000004030000000000001C1F0F0707070707")
133 DEFCHR$(&H84)=HEXCHR$("000000010100000000000000010100000000010307070703")
134 DEFCHR$(&H85)=HEXCHR$("000000000000000000000000000000000101000000000000")
135 DEFCHR$(&H86)=HEXCHR$("000000000000000001000000000000010000000000070F1F")
136 DEFCHR$(&H87)=HEXCHR$("010000000000000001000000000000001F1F1F1F1F1F1F1F")
137 DEFCHR$(&H88)=HEXCHR$("0000000304040300010300030404030001031F3F7C7C7F7F")
138 DEFCHR$(&H89)=HEXCHR$("00000000000000000000000000000000FFFF7F1F06000000")
139 DEFCHR$(&HA0)=HEXCHR$("00000000000000000A152A458A55AA550000000000000000")
140 DEFCHR$(&HA1)=HEXCHR$("0000000000000000AA1188148A4120150000000000000000")
141 DEFCHR$(&HA2)=HEXCHR$("0A152A458A55AA5500000000000000000000000000000000")
142 DEFCHR$(&HA3)=HEXCHR$("AA1188148A41201500000000000000000000000000000000")
143 DEFCHR$(&HAC)=HEXCHR$("00000000000000000000008142A048A45AA55000000000000")
144 DEFCHR$(&HAE)=HEXCHR$("000000000000000000000000000854AA550000000000000000")
145 DEFCHR$(&H90)=HEXCHR$("0C12130F1B07071E6CD2F37FE7F8F8200C12130F1B07071E")
146 DEFCHR$(&H91)=HEXCHR$("0C12130F030000008C12130F030000008CF2F37FFFFCFCF8")
147 DEFCHR$(&H92)=HEXCHR$("000000000000008000000000000000800000000000E0F0F8")
148 DEFCHR$(&H93)=HEXCHR$("80000000000000008000000000000000F8F8F8F8F8F8F8F8")
149 DEFCHR$(&H94)=HEXCHR$("000000C02020C00080C000C02020C00080C0F8FC3E3EFEFE")
150 DEFCHR$(&H95)=HEXCHR$("00000000000000000000000000000000FFFFFEF860000000")
151 DEFCHR$(&H96)=HEXCHR$("000000000000C02000000040C000C02000000040C0E0F038")
152 DEFCHR$(&H97)=HEXCHR$("20C000000000000020C000000000000038F8F0E0E0E0E0E0")
153 DEFCHR$(&H98)=HEXCHR$("000000008080000000000000808000000000080C0E0E0E0C0")
154 DEFCHR$(&H99)=HEXCHR$("000000000000000000000000000000008080000000000000")
155 DEFCHR$(&HB0)=HEXCHR$("0000000000000000A854AA55AA55AA550050081402040204")
156 DEFCHR$(&HB1)=HEXCHR$("0000000000000000AA55AA152A5488500000000000000000")
157 DEFCHR$(&HB2)=HEXCHR$("A854AA55AA55AA5500000000000000000050081402040204")
158 DEFCHR$(&HB3)=HEXCHR$("AA55AA152A54885000000000000000000000000000000000")
159 DEFCHR$(&HAD)=HEXCHR$("0000000000000000A85428512245AA550010080002040000")
160 DEFCHR$(&HAF)=HEXCHR$("0000000000000000000000000050AA550000000000100200")
161 DEFCHR$(&HAA)=HEXCHR$("0000000000000000071F1737373F1F1E071F1737363F1F1E")
162 DEFCHR$(&H8B)=HEXCHR$("FFFFFF7F7F3F1F07FFFFFF7F7F3F1F07FFFFFF7F7F3F1F07")
163 DEFCHR$(&H8C)=HEXCHR$("030F1F3E116377BF030F1F3E136377BF030F1F3E13677FBF")
164 DEFCHR$(&H8D)=HEXCHR$("C5F37F7B3813170FC5F37F7B3813170FDDFF7F7F3F17170F")
165 DEFCHR$(&H8E)=HEXCHR$("03041631668C63AD03041633678C63AD03041633678F6FBD")
166 DEFCHR$(&H8F)=HEXCHR$("96632ED5704A160396632ED5704A16039E7F2FD7774F1707")
167 DEFCHR$(&H42)=HEXCHR$("0210004008800304021000420B800304021000420B870F1C")
168 DEFCHR$(&H43)=HEXCHR$("240380010840000824038001084000083C1F8F070F47070F")
169 DEFCHR$(&H40)=HEXCHR$("000000000200000000000000020000000000000002010303")
170 DEFCHR$(&H41)=HEXCHR$("000000000000000000000000000000000303010103030707")
171 DEFCHR$(&H8A)=HEXCHR$("000000000060F3E7000000020360F3E7000000020367FFFF")
172 DEFCHR$(&H3E)=HEXCHR$("60030000000000006003000000000000781F0F0707070707")
173 DEFCHR$(&HA4)=HEXCHR$("000000000000000008142A048A44A0510000000000000000")
174 DEFCHR$(&HA5)=HEXCHR$("0000000000000000A20108148A4000150000000000000000")
175 DEFCHR$(&HA6)=HEXCHR$("000000000000000000000008142A04000000000000000000")
176 DEFCHR$(&HA7)=HEXCHR$("00000000000000008A44A051A201AA550000000000000000")
177 DEFCHR$(&HAB)=HEXCHR$("0000000000000000E0F8E8ECECFCF878E0F8E8EC6CFCF878")
178 DEFCHR$(&H9B)=HEXCHR$("FFFFFFFEFEFCF8E0FFFFFFFEFEFCF8E0FFFFFFFEFEFCF8F8")
179 DEFCHR$(&H9C)=HEXCHR$("80687C5E8E7338E580687C5E8E7338E580687C5E8EF3F8FD")
180 DEFCHR$(&H9D)=HEXCHR$("B663B31FCEEEDCB8B663B31FCEEEDCB8FEFBFBFFFEFEFCF8")
181 DEFCHR$(&H9E)=HEXCHR$("60101844301B4EA860101844301B4EA86010184430FBFEF8")
182 DEFCHR$(&H9F)=HEXCHR$("8D4421B246CC5C308D4421B246CC5C30FDFCF9FAFEFCFCF8")
183 DEFCHR$(&H52)=HEXCHR$("100280001140008410028000114000841002800011E0F0FC")
184 DEFCHR$(&H53)=HEXCHR$("90010010020084209001001002008420F8F9F8F8FAF8FCF8")
185 DEFCHR$(&H50)=HEXCHR$("0000000000000006000000000040C0060000000004ECF8FC")
186 DEFCHR$(&H51)=HEXCHR$("70B040000000000070B04000000000007CBEDEFCFCF8F8F8")
```

```
187 DEFCHR$(&H9A)=HEXCHR$("0000000000000080000000000000008000000000000E0F0F8")
188 DEFCHR$(&H3F)=HEXCHR$("0000000000000000000000000000000078F8F8F8F8F8F8F8")
189 DEFCHR$(&HB4)=HEXCHR$("0000000000000000A854285122058A450050081002040104")
190 DEFCHR$(&HB5)=HEXCHR$("0000000000000000A0400A05224400100000000000000000")
191 DEFCHR$(&HB6)=HEXCHR$("000000000000000000000000A85428510000000000500800")
192 DEFCHR$(&HB7)=HEXCHR$("000000000000000022058A45A0402A550204020000000000")
193 DEFCHR$(&HC0)=HEXCHR$("000000000000000000000000030F1F1F00000000030F1F1F")
194 DEFCHR$(&HC1)=HEXCHR$("0000000000000000191717191F1F3F1C191717191F1F3F1C")
195 DEFCHR$(&HC2)=HEXCHR$("0000000000000000000000030F1F1F1F000000030F1F1F1F")
196 DEFCHR$(&HC3)=HEXCHR$("00000000000000001F1B1717180F03011F1B1717180F0301")
197 DEFCHR$(&HC4)=HEXCHR$("000000000000000000000000030F1F1700000000030F1F17")
198 DEFCHR$(&HC5)=HEXCHR$("000000000000000017373F3F1F1F1F1E17363F3F1F1F1F1E")
199 DEFCHR$(&HC6)=HEXCHR$("0000000000000000000000030F1F1717000000030F1F1717")
200 DEFCHR$(&HC7)=HEXCHR$("0000000000000000373F3F1F1F0F0100363F3F1F1F0F0100")
201 DEFCHR$(&HC8)=HEXCHR$("0000000000000000000000030F1F3737000000030F1F3737")
202 DEFCHR$(&HC9)=HEXCHR$("0000000000000000371F1F1F1F0F0F07361F1F1F1F0F0F07")
203 DEFCHR$(&HCA)=HEXCHR$("000000000000000000000000030F1F1F00000000030F1F1F")
204 DEFCHR$(&HCB)=HEXCHR$("00000000000000001F3F3F3F1F1F1F1E1F3F3F3F1F1F1F1E")
205 DEFCHR$(&HCC)=HEXCHR$("0000000000000000000000030F1F1F1F000000030F1F1F1F")
206 DEFCHR$(&HCD)=HEXCHR$("00000000000000003F3F3F1F1F0F01003F3F3F1F1F0F0100")
207 DEFCHR$(&HCE)=HEXCHR$("0000000000000000000000030F1F3F3F000000030F1F3F3F")
208 DEFCHR$(&HCF)=HEXCHR$("00000000000000003F1F1F1F1F0F0F073F1F1F1F1F0F0F07")
209 DEFCHR$(&HD0)=HEXCHR$("000000000000000000000000E0F8E8EC00000000E0F8E8E8")
210 DEFCHR$(&HD1)=HEXCHR$("0000000000000000ECF8F8F8F8F8FC38ECF8F8F8F8F8FC38")
211 DEFCHR$(&HD2)=HEXCHR$("0000000000000000000000E0F8E8ECEC000000E0F8E8E8EC")
212 DEFCHR$(&HD3)=HEXCHR$("0000000000000000F8F8F8F8F8F0C0E0F8F8F8F8F8F0C0E0")
213 DEFCHR$(&HD4)=HEXCHR$("000000000000000000000000C0F0F8E800000000C0F0F8E8")
214 DEFCHR$(&HD5)=HEXCHR$("0000000000000000E8ECFCFCF8F8F878E86CFCFCF8F8F878")
215 DEFCHR$(&HD6)=HEXCHR$("0000000000000000000000C0F0F8ECEC000000C0F0F8ECEC")
216 DEFCHR$(&HD7)=HEXCHR$("0000000000000000ECF8F8F8F8F0F0E06CF8F8F8F8F0F0E0")
217 DEFCHR$(&HD8)=HEXCHR$("0000000000000000000000C0F0F8E8E8000000C0F0F8E8E8")
218 DEFCHR$(&HD9)=HEXCHR$("0000000000000000ECFCFCF8F8F080006CFCFCF8F8F08000")
219 DEFCHR$(&HDA)=HEXCHR$("000000000000000000000000C0F0F8F800000000C0F0F8F8")
220 DEFCHR$(&HDB)=HEXCHR$("0000000000000000F8FCFCFCF8F8F878F8FCFCFCF8F8F878")
221 DEFCHR$(&HDC)=HEXCHR$("0000000000000000000000C0F0F8FCFC000000C0F0F8FCFC")
222 DEFCHR$(&HDD)=HEXCHR$("0000000000000000FCF8F8F8F8F0F0E0FCF8F8F8F8F0F0E0")
223 DEFCHR$(&HDE)=HEXCHR$("0000000000000000000000C0F0F8F8F8000000C0F0F8F8F8")
224 DEFCHR$(&HDF)=HEXCHR$("0000000000000000FCFCFCF8F8F08000FCFCFCF8F8F08000")
225 DEFCHR$(&HE0)=HEXCHR$("00000000000000000000000107050D0D000000010705050D")
226 DEFCHR$(&HE1)=HEXCHR$("0000000000000000F0F0F070F0F07000F0F0F070F0F07000")
227 DEFCHR$(&HE2)=HEXCHR$("0000000000000000000000001F7F5FDF000000001F7F5F5F")
228 DEFCHR$(&HE3)=HEXCHR$("0000000000000000DE7FFFFFFF3F3F1CDE7FFFFFFF3F3F1C")
229 DEFCHR$(&HE4)=HEXCHR$("00000000000000000000000F3F7F7FFF0000000F3F7F7F7F")
230 DEFCHR$(&HE5)=HEXCHR$("0000000000000000FFFFFFFF7F3F3C1C7FFFFFFF7F3F3C1C")
231 DEFCHR$(&HE6)=HEXCHR$("000000000000000000000000000307070000000000030707")
232 DEFCHR$(&HE7)=HEXCHR$("000000000000000006050506070703000605050607070300")
233 DEFCHR$(&HE8)=HEXCHR$("000000000000000000000000071F7FEF00000000071F3F2F")
234 DEFCHR$(&HE9)=HEXCHR$("0000000000000000EF6F7F3F3E3D1D1E6F6C7F3F3E3D1D1E")
235 DEFCHR$(&HEA)=HEXCHR$("000000000000000000000000000001070000000000000107")
236 DEFCHR$(&HEB)=HEXCHR$("0000000000000000F0B0B1B1F1F0E000F0B0B1B1F1F0E000")
237 DEFCHR$(&HEC)=HEXCHR$("00000000000000000000000001070F1F0000000001070F1F")
238 DEFCHR$(&HED)=HEXCHR$("00000000000000001F1F0F0F0F0F07001F1F0F0F0F0F0700")
239 DEFCHR$(&HEE)=HEXCHR$("000000000000000000000000071F3F3F00000000071F3F3F")
240 DEFCHR$(&HEF)=HEXCHR$("00000000000000003F7F7F7F3F3F3F1E3F7F7F7F3F3F3F1E")
241 DEFCHR$(&HF0)=HEXCHR$("0000000000000000000000F0FCFEFEFF000000F0FCFEFEFE")
242 DEFCHR$(&HF1)=HEXCHR$("0000000000000000FFFFFFFFFEFC3C38FEFFFFFFFEFC3C38")
243 DEFCHR$(&HF2)=HEXCHR$("00000000000000000000000000C0E0E00000000000C0E0E0")
244 DEFCHR$(&HF3)=HEXCHR$("000000000000000060A0A060E0E0C00060A0A060E0E0C000")
245 DEFCHR$(&HF4)=HEXCHR$("00000000000000000000000080E0A0B0B0000080E0A0A0B0")
246 DEFCHR$(&HF5)=HEXCHR$("0000000000000000F0F0F0E0F0F0E000F0F0F0E0F0F0E000")
247 DEFCHR$(&HF6)=HEXCHR$("000000000000000000000000F8FEFAFB00000000F8FEFAFA")
248 DEFCHR$(&HF7)=HEXCHR$("00000000000000007BFEFFFFFFFCFC387BFEFFFFFFFCFC38")
249 DEFCHR$(&HF8)=HEXCHR$("00000000000000000000000080E0F0D00000000080E0F0D0")
250 DEFCHR$(&HF9)=HEXCHR$("0000000000000000D0D0F8F8F8F0E000D0D0F8F8F8F0E000")
251 DEFCHR$(&HFA)=HEXCHR$("0000000000000000000000000000E0F8000000000000E0F8")
252 DEFCHR$(&HFB)=HEXCHR$("0000000000000000FCF6F6F6FCFCFC78FCF6F636FCFCFC78")
253 DEFCHR$(&HFC)=HEXCHR$("000000000000000000000000E0F8FCFC00000000E0F8FCFC")
254 DEFCHR$(&HFD)=HEXCHR$("0000000000000000FFFFFFFEFEFC7C78F8F8FEFEFEFC7C78")
255 DEFCHR$(&HFE)=HEXCHR$("000000000000000000000000B0F8F8F800000000B0F8F8F8")
256 DEFCHR$(&HFF)=HEXCHR$("0000000000000000F8F0F0F0F0F0E000F8F0F0F0F0F0E000")
2000 END
```

メインプログラムリスト

```
5 CLR
10 DIM EX(9),EY(9),EM(9),EMF(9),EPTAD(20),EPN(9),ETCT(9),ESLP(9)
20 DIM UX(9),UY(9),UM(9),UMF(9),UPTAD(10),UPN(9),UTCT(9),USLP(9)
30 DIM MX(9),MY(9),MM(9),MMF(19),FLY(9),MTM(9)
35 DIM SX(19),SY(19),SM(19),SMF(19),BFG(19),BCT(19),STPTAD(4)
40 DIM FPTAD(20)
50 REM
60 REM EX ... ｴﾋﾞｰﾗ X
70 REM UX ... ﾕﾆｺｰﾝ X
80 REM MX ... ｷﾉｺ X
90 REM SX ... ｲﾜ X
100 REM
110 REM ﾒｲﾝ
120 REM
125 WIDTH 40,25:CONSOLE ,,0,1:CLS 0
130 MAZEAD=&H6000:PATAD=MAZEAD+600:VRMWK=&H9000:GOSUB 20000:GOSUB 19500
140 GOSUB 5170:REM ﾊﾟﾀｰﾝ ｱﾄﾞﾚｽ ｾｯﾄ
500 REM
510 REM INIT 1
520 REM
530  SCORE=0:SIDE=0:FKAZU=5
600 REM
610 REM INIT 2
620 REM
630  SIDE=SIDE+1:NSFG=0
635  FX=1:FY=4:FM=3:FMF=0:FPN=0:DYFG=0:KCNT=0:KFG=0:FTCT=0:FDCT=0:XL=0:TM=1500
636  GOSUB 19000:REM ﾍﾝｽｳ ｸﾘｱ
638  FOR I=VRMWK TO VRMWK+999:POKE I,&H20:NEXT:CLS 0
640  GOSUB 1000:GOSUB 1060:GOSUB5000:GOSUB14000
700 REM
710 REM MOVE
720 REM
730  GOSUB 3000:REM ﾕﾆｺｰﾝ ｲﾄﾞｳ
740  GOSUB 8000:REM ｲﾜ ﾉ ｲﾄﾞｳ
745  IF NSFG THEN 630
750  GOSUB 4000:REM ｴﾋﾞｰﾗ ｲﾄﾞｳ
760  GOSUB 6040:REM ﾌﾗｯﾋﾟｰ ｲﾄﾞｳ
762  IF FKAZU=0 THEN GOSUB 19800:GOTO 500
765  IF DYFG THEN 635
770  GOSUB 8000:REM ｲﾜ ﾉ ｲﾄﾞｳ
775  IF NSFG THEN 630
780  GOSUB 10000:REM ｷﾉｺ ﾉ ｲﾄﾞｳ
785  IF TM THEN XL=XL XOR 1:TM=TM-XL:GOSUB15530 ELSE IF FDCT=0 THEN FDCT=10
786  K$=INKEY$(0):IF K$="G" OR K$="g" THEN FKAZU=FKAZU-1:GOTO635
790 GOTO 700
800 REM
1000 CLS 0
1010 Y=0:GOSUB 1040:X=0:GOSUB 1050:X=39:GOSUB1050:Y=3:GOSUB 1040
1020 Y=24:GOSUB 1040
1025 LOCATE13,1:COLOR 7,2:GPRINT"&'()*":LOCATE29,2:COLOR 7,4:GPRINT"-.<%&(+:":LO
CATE28,1:COLOR 7,6:GPRINT":$";CHR$(&H22);"*":LOCATE11,1:GPRINT"ｨ":LOCATE26,1:GPR
INT"ｨ":LOCATE11,2:GPRINT"ｨ":LOCATE26,2:GPRINT"ｨ":GOSUB15500
1030 RETURN
1040 LOCATE0,Y:FOR X=0 TO 39:GPRINT CHR$(&HA8);:POKE X+Y*40+VRMWK,&HA8:NEXT:RETU
RN
1050 FOR Y=1 TO 23:LOCATEX,Y:GPRINT CHR$(&HA8):POKE X+Y*40+VRMWK,&HA8:NEXT:RETUR
N
1060 REM
1070 REM BLOCK WRITE
1080 REM
2000 Y=4:MZAD=MAZEAD:CT=(SIDE-1) MOD 4
2010 IF CT THEN MZAD=MZAD+PEEK(MZAD):CT=CT-1:GOTO 2010
2020  SPNT=0:MPNT=SPNT:EPNT=SPNT:UPNT=SPNT
2030 FOR D=0 TO 9
2040  X=1
2050  FOR C=0 TO 2
2060   MZAD=MZAD+1
2070   B=PEEK(MZAD)
2080    FOR A=7 TO 0 STEP -1
2090     IF X>37 THEN 2120
```

```
2100     IF B AND 2^A THEN GPUT (X*2,Y*8)-PATAD:GOSUB 2300
2110     X=X+2
2120    NEXT
2130  NEXT
2140  A=B AND &H1F:IF A=0 THEN 2210
2150  FOR C=1 TO A
2160   MZAD=MZAD+1:B=PEEK(MZAD) AND &HC0:X=(PEEK(MZAD) AND &H3F)
2170   IF B=&H40 THEN GOSUB 2260:MPNT=MPNT+1:GOTO2200
2180   IF B=&H80 THEN GOSUB 2270:EPNT=EPNT+1:GOTO2200
2190   IF B=&HC0 THEN GOSUB 2280:UPNT=UPNT+1  ELSE GOSUB 2290:SPNT=SPNT+1
2200  NEXT
2210  Y=Y+2
2220 NEXT
2230 MZAD=MZAD+1:X=PEEK(MZAD):Y=PEEK(MZAD+1):GOSUB 2290:BFG(SPNT)=1:SPNT=SPNT+1
2240 MZAD=MZAD+2:BWX=PEEK(MZAD):BWY=PEEK(MZAD+1):GPUT (BWX*2,BWY*8)-PATAD+6
2250 RETURN
2260 MX(MPNT)=X:MY(MPNT)=Y+1:MM(MPNT)=3:FLY(MPNT)=1:RETURN
2270 EX(EPNT)=X:EY(EPNT)=Y:EM(EPNT)=3:EMF(EPNT)=0:EPN(EPNT)=0:RETURN
2280 UX(UPNT)=X:UY(UPNT)=Y:UM(UPNT)=0:UMF(UPNT)=0:UPN(UPNT)=0:RETURN
2290 SX(SPNT)=X:SY(SPNT)=Y:SM(SPNT)=0:SMF(SPNT)=0:BFG(SPNT)=0:BCT(SPNT)=0:RETURN
2300 PKAD=VRMWK+X+Y*40
2310 POKE PKAD,&HA8:POKE PKAD+1,&HA8:POKE PKAD+40,&HA8:POKE PKAD+41,&HA8:RETURN
2320 REM
2330 REM ﾕﾆｺｰﾝ ｲﾄﾞｳﾞﾙｰﾁﾝ
2340 REM
3000 FOR I=0 TO UPNT-1
3010  IF UX(I)=0 THEN 3129
3020  IF UMF(I) THEN GOSUB 3260:GOTO 3120
3025  IF UTCT(I) THEN GOSUB 3500:GOTO3120
3027  IF USLP(I) THEN GOSUB 3600:GOTO3120
3030  IF RND(10) THEN 3070
3040  SA=UX(I)-FX
3050  IF ABS(SA)>10 OR UY(I)<>FY THEN UM(I)=RND(2)*2:GOTO3070
3060  IF SA>0 THEN UM(I)=0 ELSE UM(I)=2
3070  SCY=UY(I)
3080  IF UM(I) THEN SCX=UX(I)+2:NX=UX(I)+1 ELSE SCX=UX(I)-1:NX=SCX
3090  S=PEEK(SCY*40+SCX+VRMWK)
3100  IF S<>&H20 AND S<>&H21 AND S<&HB8 THEN 3120
3110  GOSUB 3140
3120  REM ﾌﾗｯﾋﾟｰ ﾆ ｻﾜｯﾀｶ ?
3121   IF FDCT THEN 3129
3122   SAX=UX(I)-FX:SAY=UY(I)-FY
3123   IF (ABS(SAX)<2) AND (ABS(SAY)<2) THEN FDCT=10
3129 NEXT
3130 RETURN
3140 REM
3150 REM ﾕﾆｺｰﾝ PUT SUB.
3160 IF UM(I) THEN ZX=1 ELSE ZX=-1
3170 GPUT(UX(I)*2,UY(I)*8)-PATAD+356:REM NULL
3180 X=UX(I):Y=UY(I):AD=PATAD+356:GOSUB 3230:REM ﾜｰｸ PUT
3190 GPUT(UX(I)*2+ZX,UY(I)*8)-UPTAD(UM(I)+UPN(I))
3200 X=NX:Y=UY(I):AD=UPTAD(UM(I)+UPN(I)):GOSUB3230:REM ﾜｰｸ PUT
3210 UX(I)=NX:UPN(I)=UPN(I) XOR 1:UMF(I)=1
3220 RETURN
3230 REM
3240 REM ﾊﾟﾀｰﾝ ﾜｰｸ PUT
3250 AD=AD+2:PKAD=VRMWK+X+Y*40:POKE PKAD,PEEK(AD):POKE PKAD+1,PEEK(AD+1):POKE PK
AD+40,PEEK(AD+2):POKE PKAD+41,PEEK(AD+3):RETURN
3255 AD=AD+2:PKAD=VRMWK+X+Y*40:POKE PKAD,PEEK(AD):POKE PKAD+1,PEEK(AD+1):RETURN
3260 REM
3270 REM ﾕﾆｺｰﾝ PUT SUB.2
3280 IF UM(I) THEN ZX=-1 ELSE ZX=1
3290 GPUT(UX(I)*2+ZX,UY(I)*8)-PATAD+356:REM NULL
3300 GPUT(UX(I)*2,UY(I)*8)-UPTAD(UM(I)+UPN(I))
3310 UPN(I)=UPN(I) XOR 1:UMF(I)=0
3320 RETURN
3470 REM
3480 REM ﾕﾆｺｰﾝ ﾂﾌﾞｽ ﾙｰﾁﾝ
3500  IF UTCT(I)=3 THEN SOUND 700,200,3:X=UX(I):Y=UY(I):GPUT(X*2,Y*8)-PATAD+356:
GPUT(X*2,Y*8+8)-UPTAD(10):AD=PATAD+366:GOSUB 3250:UTCT(I)=UTCT(I)-1:RETURN
```

```
3510  IF UTCT(I)=1 THEN X=UX(I):Y=UY(I):GPUT(X*2,Y*8+8)-PATAD+362:Y=Y+1:AD=PATAD
+362:GOSUB 3255:UX(I)=0:SCORE=SCORE+5:GOSUB 15500
3520  UTCT(I)=UTCT(I)-1:RETURN
3580 REM
3590 REM ﾕﾆｺｰﾝ ﾈﾙ
3600 IF USLP(I)>146 THEN GPUT(UX(I)*2,UY(I)*8)-UPTAD(USLP(I)-141):USLP(I)=USLP(I
)-1:RETURN
3610 IF USLP(I)>124 THEN GPUT(UX(I)*2,UY(I)*8)-UPTAD(UM(I)+1):USLP(I)=USLP(I)-1:
RETURN
3620 IF USLP(I)>20 THEN IF USLP(I) AND 3 THEN A=A ELSE UPN(I)=UPN(I) XOR 1 ELSE
3630
3625 GPUT(UX(I)*2,UY(I)*8)-UPTAD(UPN(I)+4):USLP(I)=USLP(I)-1:RETURN
3630 GPUT(UX(I)*2,UY(I)*8)-UPTAD(UM(I)+1):USLP(I)=USLP(I)-1:RETURN
3970 REM
3980 REM ｴﾋﾞｰﾗ ｲﾄﾞｳ ﾙｰﾁﾝ
3990 REM
4000 FOR I=0 TO EPNT-1
4010  IF EX(I)=0 THEN 4175
4020  IF EMF(I) THEN GOSUB4190:GOTO 4170
4025  IF ETCT(I) THEN GOSUB 4600:GOTO 4170
4027  IF ESLP(I) THEN GOSUB 4800:GOTO 4170
4030  IF RND(10) THEN 4360
4040  SA=EX(I)-FX:SA2=EY(I)-FY
4050  IF ABS(SA)>5 OR ABS(SA2)>5 THEN 4340
4060  IF SA=0 THEN 4280 ELSE IF SA2=0 THEN 4310
4070  IF RND(2) THEN IF SA<0 THEN EM(I)=2:GOTO 4090 ELSE EM(I)=0:GOTO 4090
4080  IF SA2<0 THEN EM(I)=3 ELSE EM(I)=1
4090  NX=EX(I):NY=EY(I):HOU=EM(I):GOSUB 4460:REM ｽｽﾑ ﾎｳｺｳ ｼﾗﾍﾞﾙ
4100  IF (A<>&H20 AND A<>&H21 AND A<&HB8) OR (B<>&H20 AND B<>&H21 AND B<&HB8) TH
EN 4170
4110  REM ｴﾋﾞｰﾗ PUT SUB.
4120 GPUT(EX(I)*2,EY(I)*8)-PATAD+356
4130  X=EX(I):Y=EY(I):AD=PATAD+356:GOSUB 3230:REM ﾜｰｸ PUT
4140  GPUT(EX(I)*2+ZX,EY(I)*8+ZY)-EPTAD(EM(I)*2+EPN(I))
4150  X=NX:Y=NY:AD=EPTAD(EM(I)*2+EPN(I)):GOSUB3230:REM ﾜｰｸ PUT
4160  EX(I)=NX:EY(I)=NY:EPN(I)=EPN(I) XOR 1:EMF(I)=1
4170  REM ﾌﾗｯﾋﾟｰ ﾆ ｻﾜｯﾀｶ ?
4171   IF FDCT THEN 4175
4172   SAX=EX(I)-FX:SAY=EY(I)-FY
4173   IF (ABS(SAX)<2) AND (ABS(SAY)<2) THEN FDCT=10
4175 NEXT
4180 RETURN
4190  REM ｴﾋﾞｰﾗ PUT SUB.2
4200  ZX=0:ZY=ZX
4210  IF EM(I)=1 THEN ZY=4:GOTO 4240
4220  IF EM(I)=2 THEN ZX=-1:GOTO 4240
4230  IF EM(I)=3 THEN ZY=-4 ELSE ZX=1
4240 GPUT(EX(I)*2+ZX,EY(I)*8+ZY)-PATAD+356
4250  GPUT(EX(I)*2,EY(I)*8)-EPTAD(EM(I)*2+EPN(I))
4260  EPN(I)=EPN(I) XOR 1:EMF(I)=0
4270 RETURN
4280 REM Y ｾｯｷﾝ
4290 IF SA2>0 THEN EM(I)=1 ELSE EM(I)=3
4300 GOTO 4090
4310 REM X ｾｯｷﾝ
4320 IF SA>0 THEN EM(I)=0 ELSE EM(I)=2
4330 GOTO 4090
4340 REM ﾗﾝｽｳ ｾｯｷﾝ
4350 EM(I)=(EM(I)+1+RND(2)*2) AND 3:GOTO 4090
4360 REM
4370 REM ﾎｳｺｳ ｲｼﾞ
4390  NX=EX(I):NY=EY(I):HOU=EM(I):GOSUB 4460:MKBUF=EM(I)
4400  IF (A=&H20 OR A=&H21 OR A>&HB7) AND (B=&H20 OR B=&H21 OR B>&HB7) THEN 4110
4410  EM(I)=RND(2)*2+((1-(EM(I) AND 1))):NX=EX(I):NY=EY(I):HOU=EM(I):GOSUB 4460
4420  IF (A=&H20 OR A=&H21 OR A>&HB7) AND (B=&H20 OR B=&H21 OR B>&HB7) THEN 4110
4430  EM(I)=EM(I)+2 AND 3:NX=EX(I):NY=EY(I):HOU=EM(I):GOSUB 4460
4440  IF (A=&H20 OR A=&H21 OR A>&HB7) AND (B=&H20 OR B=&H21 OR B>&HB7) THEN 4110
4450  EM(I)=MKBUF+2 AND 3:GOTO 4170
4460 REM
4470 REM A,B <= VRMWK
```

```
4490  SCX=NX:SCY=NY:ZX=0:ZY=ZX
4500  IF HOU=1 THEN NY=NY-1:SCY=NY:SCX2=NX+1:SCY2=SCY:ZY=-4:GOTO 4540
4510  IF HOU=2 THEN NX=NX+1:SCX=NX+1:SCX2=SCX:SCY2=NY+1:ZX=1:GOTO 4540
4520  IF HOU=3 THEN NY=NY+1:SCY=NY+1:SCX2=NX+1:SCY2=SCY:ZY=4:GOTO 4540
4530  NX=NX-1:SCX=NX:SCX2=SCX:SCY2=NY+1:ZX=-1
4540  A=PEEK(VRMWK+SCX+SCY*40):B=PEEK(VRMWK+SCX2+SCY2*40):RETURN
4599  REM ｴﾋﾞｰﾗ ﾂﾌﾞﾚﾙ ﾙｰﾁﾝ
4600  IF ETCT(I)=3 THEN SOUND 700,200,3:X=EX(I):Y=EY(I):GPUT(X*2,Y*8)-PATAD+356:
GPUT(X*2,Y*8+8)-EPTAD(20):AD=PATAD+372:GOSUB 3250:ETCT(I)=ETCT(I)-1:RETURN
4610  IF ETCT(I)=1 THEN X=EX(I):Y=EY(I):GPUT(X*2,Y*8+8)-PATAD+362:Y=Y+1:AD=PATAD
+362:GOSUB 3255:EX(I)=0:SCORE=SCORE+10:GOSUB15500
4620  ETCT(I)=ETCT(I)-1:RETURN
4780 REM
4790 REM ｴﾋﾞｰﾗ ﾈﾙ
4800 IF ESLP(I)>146 THEN GPUT(EX(I)*2,EY(I)*8)-EPTAD(ESLP(I)-137):ESLP(I)=ESLP(I
)-1:RETURN
4810 IF ESLP(I)>124 THEN GPUT(EX(I)*2,EY(I)*8)-EPTAD(9):ESLP(I)=ESLP(I)-1:RETURN
4820 IF ESLP(I)>20 THEN IF ESLP(I) AND 3 THEN A=A ELSE EPN(I)=EPN(I) XOR 1 ELSE
4840
4830 GPUT(EX(I)*2,EY(I)*8)-EPTAD(EPN(I)+14):ESLP(I)=ESLP(I)-1:RETURN
4840 GPUT(EX(I)*2,EY(I)*8)-EPTAD(9):ESLP(I)=ESLP(I)-1:RETURN
4970 REM
4980 REM ﾊﾟﾀｰﾝ ﾋｮｳｼﾞ
4990 REM
5000 FOR D=0 TO SPNT-1
5010  IF BFG(D) THEN AD=PATAD+40:X=SX(D):Y=SY(D):GPUT(X*2,Y*8)-AD:GOSUB 3250:GOT
O 5030
5020  AD=PATAD+10:X=SX(D):Y=SY(D):GPUT(X*2,Y*8)-AD:GOSUB 3250
5030 NEXT
5040 FOR D=0 TO MPNT-1
5050  IF MPNT THEN LOCATE MX(D),MY(D):GPRINTCHR$(&H21):GOSUB5140
5060 NEXT
5070 FOR D=0 TO EPNT-1
5080  IF EPNT THEN GPUT(EX(D)*2,EY(D)*8)-PATAD+88:GOSUB 5150
5090 NEXT
5100 FOR D=0 TO UPNT-1
5110  IF UPNT THEN GPUT(UX(D)*2,UY(D)*8)-UPTAD(0):GOSUB 5160
5120 NEXT
5130 RETURN
5140 PKAD=MX(D)+MY(D)*40+VRMWK:POKE PKAD,&H21:RETURN
5150 PKAD=EX(D)+EY(D)*40+VRMWK:POKE PKAD,&H6A:POKE PKAD+1,&H7A:POKE PKAD+40,&H6B
:POKE PKAD+41,&H7B:RETURN
5160 PKAD=UX(D)+UY(D)*40+VRMWK:POKE PKAD,&H82:POKE PKAD+1,&H92:POKE PKAD+40,&H83
:POKE PKAD+41,&H93:RETURN
5170 REM
5180 REM ﾊﾟﾀｰﾝ ｱﾄﾞﾚｽ ｾｯﾄ
5190 REM
6000 FOR I=0 TO 10:UPTAD(I)=PATAD+170+(I*6):NEXT
6010 FOR I=0 TO 20:EPTAD(I)=PATAD+46+(I*6):NEXT
6020 FOR I=0 TO 19:FPTAD(I)=PATAD+234+(I*6):NEXT:FPTAD(20)=PATAD+352
6025 FOR I=0 TO 3:STPTAD(I+1)=PATAD+16+(I*6):NEXT:STPTAD(0)=PATAD+362
6030 RETURN
6040 REM
6050 REM ﾌﾗｯﾋﾟｰ ﾉ ｲﾄﾞｳ
6060 REM
7000 REM
7010 IF FMF THEN 7180
7015 IF KFG THEN 7600
7017 IF FTCT THEN 7900
7018 IF FDCT THEN 7950
7020 STK=STICK(0):STR=STRIG(0)
7040  IF STK=8 THEN HOU=1:GOTO 7080
7050  IF STK=6 THEN HOU=2:GOTO 7080
7060  IF STK=2 THEN HOU=3:GOTO 7080
7070  IF STK=4 THEN HOU=0:GOTO 7080
7075  IF STR=-1 THEN KFG=1:FPN=0:GOSUB 7610:GOSUB 7800:RETURN
7076  IF FM AND 1 THEN GPUT(FX*2,FY*8)-FPTAD(FM/2+16) ELSE GPUT(FX*2,FY*8)-FPTAD
(FM*2+1)
7077  RETURN
7080  NX=FX:NY=FY:GOSUB 4460:FM=HOU
7090  IF A<&HA0 AND B<&HA0 THEN 7120
```

```
7100  GOSUB 7270
7105  IF FMF=0 THEN NX=FX:NY=FY:ZX=0:ZY=0:AFG=1:GOSUB 7120
7110  RETURN
7120 GPUT(FX*2,FY*8)-PATAD+356
7130 GPUT(FX*2+ZX,FY*8+ZY)-FPTAD(FM*2+FPN)
7140 X=FX:Y=FY:AD=PATAD+356:GOSUB 3230
7150 X=NX:Y=NY:AD=FPTAD(FM*2+FPN):GOSUB 3230
7160 FX=NX:FY=NY:FPN=FPN XOR 1:FMF=1
7170 RETURN
7180  HOU=FM:GOSUB 8130
7210  IF AFG THEN ZX=0:ZY=0
7220 GPUT(FX*2+ZX,FY*8+ZY)-PATAD+356
7230  GPUT(FX*2,FY*8)-FPTAD(FM*2+FPN)
7240  FPN=FPN XOR 1:FMF=0:AFG=0:RETURN
7250 REM
7260 REM ｲﾜ ﾉ ﾊﾝﾀﾞﾝ
7270 NXB=NX:NYB=NY:ZXB=ZX:ZYB=ZY
7280 FOR I=0 TO SPNT-1
7290  SAX=SX(I)-FX:SAY=SY(I)-FY
7300  A=FM+1:ON A GOSUB 7340,7360,7380,7400
7310 NEXT:RETURN
7320 REM
7330 REM ｲﾜ ﾉ ｶﾞｲﾄｳ ｻﾞﾋｮｳ ｻｰﾁ
7340  IF (SAX=-2) AND (ABS(SAY)<2) THEN GOSUB 7440
7350  RETURN
7360  IF (ABS(SAX)<2) AND (SAY=-2) THEN GOSUB 7440
7370  RETURN
7380  IF (SAX=2) AND (ABS(SAY)<2) THEN GOSUB 7440
7390  RETURN
7400  IF (ABS(SAX)<2) AND (SAY=2) THEN GOSUB 7440
7410  RETURN
7420 REM
7430 REM STONE 1 BREAK
7440 C=A:D=B:NX=SX(I):NY=SY(I):HOU=FM:GOSUB 4460
7450 IF (A=&H20 OR A=&H21) AND (B=&H20 OR B=&H21) THEN IF SY(I)=FY THEN 7490 ELS
E RETURN
7455 IF C=&HA2 OR C=&HA3 OR C=&HB2 OR C=&HB3 OR C=&HA8 OR C=&HA9 OR D=&HA2 OR D=
&HA3 OR D=&HB2 OR D=&HB3 OR D=&HA8 OR D=&HA9 THEN RETURN
7460 IF BFG(I) THEN RETURN
7470 IF BCT(I) THEN RETURN ELSE BCT(I)=5:RETURN
7480 REM ｲﾜ ｳｺﾞｶｽ
7490 IF HOU AND 1 THEN RETURN
7495 GPUT(SX(I)*2,SY(I)*8)-PATAD+356
7500 X=SX(I):Y=SY(I):AD=PATAD+356:GOSUB 3230
7510 IF BFG(I) THEN AD=PATAD+40:GPUT(SX(I)*2+ZX,SY(I)*8+ZY)-AD:X=NX:Y=NY:GOSUB 3
230:GOTO 7530
7520 AD=PATAD+10:GPUT(SX(I)*2+ZX,SY(I)*8+ZY)-AD:X=NX:Y=NY:GOSUB 3230
7530 SX(I)=NX:SY(I)=NY:SM(I)=HOU:SMF(I)=1
7540 NX=NXB:NY=NYB:ZX=ZXB:ZY=ZYB:GOSUB 7120:RETURN
7550 REM
7560 REM ｷﾉｺ ﾅｹﾞﾙ
7600  KFG=0
7610  GPUT(FX*2,FY*8)-FPTAD(8+FM*2+FPN):FPN=FPN XOR 1:RETURN
7770 REM
7780 REM ｷﾉｺ ｻﾞﾋｮｳ ｾｯﾄ
7800  IF KCNT=0 THEN RETURN
7801  IF FM=1 THEN SCX=FX+1:SCY=FY-1:GOTO7804
7802  IF FM=2 THEN SCX=FX+2:SCY=FY:GOTO7804
7803  IF FM=3 THEN SCX=FX:SCY=FY+2 ELSE SCX=FX-1:SCY=FY
7804  A=PEEK(VRMWK+SCX+SCY*40):IF A>&H9F AND A<&HB8 THEN RETURN
7805  J=0
7806  IF J>MPNT-1 THEN RETURN
7807  IF MX(J) THEN J=J+1:GOTO 7806
7810  KCNT=KCNT-1:GOSUB13000:SOUND 500,100,1:MM(J)=FM:MX(J)=SCX:MY(J)=SCY:FLY(J)
=1:MTM(J)=10:RETURN
7880 REM
7890 REM ﾌﾗｯﾋﾟｰ ﾂﾌﾞｽ ﾙｰﾁﾝ
7900  IF FTCT=3 THEN SOUND 1000,900,3:X=FX:Y=FY:GPUT(X*2,Y*8)-PATAD+356:GPUT(X*2
,Y*8+8)-PATAD+384:AD=PATAD+388:GOSUB 3250:FTCT=FTCT-1:RETURN
7910  IF FTCT=1 THEN FKAZU=FKAZU-1:X=FX:Y=FY+1:AD=PATAD+362:GOSUB 3255:DYFG=1:IF
 FKAZU THEN GPUT(X*2,(Y-1)*8)-PATAD+362
```

```
7920  FTCT=FTCT-1:RETURN
7948 REM
7949 REM ﾌﾗｯﾋﾟｰ ｼﾞﾀﾊﾞﾀ
7950  IF FDCT=10 THEN X=FX:Y=FY:GPUT(X*2,Y*8)-PATAD+356:GPUT(X*2,Y*8+8)-FPTAD(19
):AD=PATAD+394:GOSUB 3250:FDCT=FDCT-1:RETURN
7951  IF FDCT=1 THEN FKAZU=FKAZU-1:X=FX:Y=FY+1:AD=PATAD+362:GOSUB 3255:DYFG=1:FD
CT=0:IF FKAZU THEN GPUT(X*2,(Y-1)*8)-PATAD+362:RETURN ELSE RETURN
7952  GPUT(FX*2,FY*8+8)-FPTAD((FDCT AND 1)+19):FDCT=FDCT-1:RETURN
7970 REM
7980 REM ｲﾜ ﾉ ｲﾄﾞｳ ﾙｰﾁﾝ
7990 REM
8000 FOR I=0 TO SPNT-1
8002  IF SX(I)=0 THEN 8025
8005  IF BCT(I) THEN GOSUB 8500:GOTO 8020
8040  IF SMF(I) THEN GOSUB 8060:GOTO 8020
8011  NX=SX(I):NY=SY(I):HOU=3:GOSUB 4490
8012  IF A=&HA8 OR A=&HA9 OR B=&HA8 OR B=&HA9 OR (A>&H9F AND A<&HB8) OR (B>&H9F
AND B<&HB8) THEN 8020
8013  IF (A=&H20 OR A=&H21) AND (B=&H20 OR B=&H21) THEN GOSUB 9200:GOTO 8025
8014  GOSUB 9000:GOTO8025:REM ｲﾜﾉ ｼﾀﾆ ﾅﾆｶ ｲﾙｶ ?
8020  IF (BFG(I)=1) AND (SX(I)=BWX) AND (BWY-SY(I)=2) THEN NSFG=1:GOSUB 15000
8025 NEXT
8030 RETURN
8040 REM
8050 REM ｲﾜ ｻﾕｳ ﾆ ｳｺﾞｶｽ
8060 HOU=SM(I):GOSUB 8130:REM ZX SET
8070 GPUT(SX(I)*2+ZX,SY(I)*8+ZY)-PATAD+356
8080 IF BFG(I) THEN GPUT(SX(I)*2,SY(I)*8)-PATAD+40:GOTO 8100:REM BLUE STONE PUT
8090 GPUT(SX(I)*2,SY(I)*8)-PATAD+10:REM RED STONE PUT
8100 SMF(I)=0:RETURN
8110 REM
8120 REM ZX,ZY SET
8130 ZX=0:ZY=ZX
8140  IF HOU=1 THEN ZY=4:GOTO 8170
8150  IF HOU=2 THEN ZX=-1:GOTO 8170
8160  IF HOU=3 THEN ZY=-4 ELSE ZX=1
8170  RETURN
8480 REM
8490 REM ｲﾜ ｸｽﾞﾚﾙ
8500  IF BCT(I)=5 THEN BCT(I)=BCT(I)-1:GPUT(SX(I)*2,SY(I)*8)-STPTAD(BCT(I)):RETU
RN
8510  IF BCT(I)=4 THEN BCT(I)=BCT(I)-1:GPUT(SX(I)*2,SY(I)*8)-STPTAD(BCT(I)):RETU
RN
8520  IF BCT(I)=3 THEN BCT(I)=BCT(I)-1:GPUT(SX(I)*2,SY(I)*8)-STPTAD(BCT(I)):X=SX
(I):Y=SY(I):AD=STPTAD(BCT(I)):GOSUB 3255:RETURN
8530  IF BCT(I)=2 THEN BCT(I)=BCT(I)-1:GPUT(SX(I)*2,SY(I)*8+8)-STPTAD(BCT(I)):RE
TURN
8540  IF BCT(I)=1 THEN BCT(I)=BCT(I)-1:GPUT(SX(I)*2,SY(I)*8+8)-PATAD+362:X=SX(I)
:Y=SY(I)+1:AD=PATAD+362:GOSUB 3255:SX(I)=0:RETURN
8980 REM
8990 REM ｲﾜﾉ ｼﾀﾆ ﾅﾆｶ ｲﾀﾗ ﾂﾌﾞｽ
9000 FOR J=0 TO EPNT-1
9010  SAX=EX(J)-SX(I):SAY=EY(J)-SY(I)
9020  IF (ABS(SAX)<2) AND (SAY=2) AND (ETCT(J)=0) THEN ETCT(J)=3
9030 NEXT
9040 FOR J=0 TO UPNT-1
9050  SAX=UX(J)-SX(I):SAY=UY(J)-SY(I)
9060  IF (ABS(SAX)<2) AND (SAY=2) AND (UTCT(J)=0) THEN UTCT(J)=3
9070 NEXT
9080 SAX=FX-SX(I):SAY=FY-SY(I)
9090 IF (ABS(SAX)<2) AND (SAY=2) AND (FTCT=0) THEN FTCT=3
9100 RETURN
9190 REM
9200 REM ｲﾜ ｵﾄｽ
9210 GPUT(SX(I)*2,SY(I)*8)-PATAD+356
9220 X=SX(I):Y=SY(I):AD=PATAD+356:GOSUB 3230
9230 IF BFG(I) THEN AD=PATAD+40:GPUT(SX(I)*2+ZX,SY(I)*8+ZY)-AD:X=NX:Y=NY:GOSUB 3
230:GOTO 9250
9240 AD=PATAD+10:GPUT(SX(I)*2+ZX,SY(I)*8+ZY)-AD:X=NX:Y=NY:GOSUB 3230
9250 SX(I)=NX:SY(I)=NY:SM(I)=HOU:SMF(I)=1:RETURN
```

```
9970 REM
9980 REM ｷﾉｺ ﾉ ｲﾄﾞｳ ﾙｰﾁﾝ
10000 REM
10010 FOR I=0 TO MPNT-1
10020  IF MX(I)=0 THEN 12000
10030  SCX=MX(I):SCY=MY(I)
10040  A=PEEK(VRMWK+SCX+SCY*40)
10050  IF A>&H9F AND A<&HB8 THEN MX(I)=0:GOTO 12000
10060  REM ｷﾉｺ ﾃｷﾆ ｱﾀｯﾀｶ ?
10065  IF FLY(I)=0 THEN 10150
10070  FOR J=0 TO UPNT-1
10080   IF UX(J)=0 THEN 10100
10090   SAX=MX(I)-UX(J):SAY=MY(I)-UY(J):IF (SAX=0 OR SAX=1) AND (SAY=0 OR SAY=1)
 THEN USLP(J)=150:MX(I)=0
10100  NEXT
10110  FOR J=0 TO EPNT-1
10120   IF EX(J)=0 THEN 10140
10130   SAX=MX(I)-EX(J):SAY=MY(I)-EY(J):IF (SAX=0 OR SAX=1) AND (SAY=0 OR SAY=1)
 THEN ESLP(J)=150:MX(I)=0
10140  NEXT
10150  SAX=MX(I)-FX:SAY=MY(I)-FY:IF (SAX=0 OR SAX=1) AND (SAY=0 OR SAY=1) THEN K
CNT=KCNT+1:GOSUB13000:MX(I)=0
10160  IF MX(I)=0 THEN 12000
10170  IF FLY(I)=0 THEN SCX=MX(I):SCY=MY(I):GOTO 11550
10180  IF MTM(I) THEN MTM(I)=MTM(I)-1 ELSE MM(I)=3
11000  IF MM(I)=1 THEN SCY=MY(I)-1:GOTO 11500
11010  IF MM(I)=2 THEN SCX=MX(I)+1:GOTO 11500
11020  IF MM(I)=3 THEN SCY=MY(I)+1 ELSE SCX=MX(I)-1
11500  A=PEEK(VRMWK+SCX+SCY*40)
11510  IF A>&H9F AND A<&HB8 THEN IF MM(I)=3 THEN FLY(I)=0:GOTO 12000 ELSE MM(I)=
3:GOTO 12000
11520  IF A<>&H20 THEN 11600
11530  GPUT(MX(I)*2,MY(I)*8)-PATAD+381:REM ｷﾉｺ ｹｽ
11540  POKE VRMWK+MX(I)+MY(I)*40,&H20:MX(I)=SCX:MY(I)=SCY
11550  GPUT(SCX*2,SCY*8)-PATAD+378:REM ｷﾉｺ ｶｸ
11560  POKE VRMWK+SCX+SCY*40,&H21
11570  GOTO 12000
11600  GPUT(MX(I)*2,MY(I)*8)-PATAD+381:REM ｷﾉｺ ｹｽ
11610  POKE VRMWK+MX(I)+MY(I)*40,&H20:MX(I)=SCX:MY(I)=SCY
12000 NEXT
12010 RETURN
13000 REM
13010 REM ﾃﾓﾁ ｷﾉｺ ｶｸ
13020 REM
13030 GPUT(26,16)-PATAD+400
13040 IF KCNT=0 THEN RETURN
13050 FOR Z=0 TO KCNT-1:GPUT(26+Z*2,16)-PATAD+378:NEXT
13060 RETURN
13070 REM
14000 REM
14010 REM ﾉｺﾘ ﾌﾗｯﾋﾟｰ ﾋｮｳｼﾞ
14020 REM
14030 FOR Z=0 TO 4:GPUT(2+Z*4,8)-PATAD+356:NEXT
14040 IF FKAZU<2 THEN RETURN
14050 FOR Z=0 TO FKAZU-2:GPUT(2+Z*4,8)-PATAD+336:NEXT
14060 RETURN
14070 REM
15000 REM
15010 REM ﾎﾞｰﾅｽ ｶｻﾝ
15020 REM
15030 IF TM>10 THEN SOUND 300,10,3:SCORE=SCORE+1:TM=TM-10:GOSUB15500:GOTO15030
15040 TM=0:GOSUB15500:WAIT 200
15050 IF KCNT THEN SOUND 900,30,3:SCORE=SCORE+3:KCNT=KCNT-1:WAIT 100:GOSUB13000:
GOSUB15500:GOTO15050
15060 GOSUB15500:GOSUB13000:WAIT 400
15070 RETURN
15500 REM
15510 REM ﾀｲﾑ & ｽｺｱ ﾋｮｳｼﾞ
15520 LOCATE19,1:SC$=RIGHT$("00000"+STR$(SCORE*10),6):COLOR 7,5:GPRINT SC$
15530 LOCATE33,1:TM$=RIGHT$("00000"+STR$(TM),5):COLOR 7,6:GPRINT TM$
16000 RETURN
```

```
19000 REM
19010 REM ﾊｲﾚﾂ ｸﾘｱ
19020 REM
19030 FOR I=0 TO 9:EX(I)=0:EY(I)=0:EM(I)=0:EMF(I)=0:EPN(I)=0:ETCT(I)=0:ESLP(I)=0
:NEXT
19040 FOR I=0 TO 9:UX(I)=0:UY(I)=0:UM(I)=0:UMF(I)=0:UPN(I)=0:UTCT(I)=0:USLP(I)=0
:NEXT
19050 FOR I=0 TO 9:MX(I)=0:MY(I)=0:MM(I)=3:MMF(I)=0:FLY(I)=0:MTM(I)=0:NEXT
19060 FOR I=0 TO 9:SX(I)=0:SY(I)=0:SM(I)=3:SMF(I)=0:BFG(I)=0:BCT(I)=0:NEXT
19070 RETURN
19080 REM
19500 REM
19510 REM 4 CHR.ERASE SUB.
19520 REM
19530 FOR I=0 TO 81:READ A$:A=VAL("&H"+A$):POKE &H9400+I,A:NEXT:RETURN
19540 REM
19550 DATA 23,23,29,09,06,00,4F,CB,7F,28,02,06,FF,26,C0,EB
19560 DATA 29,29,29,09,29,29,29,29,44,4D,29,29,09,19,01,00
19570 DATA 00,11,50,00,22,3E,94,D3,5C,CD,39,94,D3,5D,CD,39
19580 DATA 94,D3,5E,CD,39,94,D3,5F,C9,ED,73,4F,94,21,00,00
19590 DATA 3E,10,32,4A,94,F9,C5,C5,19,3E,00,3D,20,F4,31,00
19600 DATA 00,C9
19610 REM
19800 REM
19810 REM GAME OVER
19820 REM
19840 IF STRIG(0) THEN RETURN ELSE 19840
19850 REM
20000 REM
20010 REM  ﾌﾞﾛｯｸ ﾃﾞｰﾀ
20020 REM
20025 REM  ** SIDE 1 **
20026 REM
20030 MEM MAZEAD,43
20040 MEM #,0,0,0,0,0,0,0,0,1,&H11
20050 MEM #,&H44,&HF3,&H80,&H64,&H80,&H41,&H1B,&H54,&HE0,2
20060 MEM #,&H1B,&HA1,&H4C,&H84,&H40,&H44,&HF3,&H80,0,0
20070 MEM #,1,&HDE,1,0,&H23,&HD3,&HD7,&HE1,17,6
20080 MEM #,37,22
20090 REM
20095 REM  ** SIDE 2 **
20096 REM
20100 MEM #,&H33
20110 MEM #,0,0,4,7,&H11,&H15,&H93,&H71,&HB1,&HC0
20120 MEM #,3,&HB8,2,3,&H23,&H67,&HBC,&HC0,7,&HBC
20130 MEM #,2,3,&H23,&H40,0,&H41,&HD3,7,&HBC,2
20140 MEM #,3,&H23,&H67,&HBC,&HC0,3,&HB8,5,&H47,&H45
20150 MEM #,&H5F,&H61,&H63,&H71,&HB1,&HC0,&H1F,4,&H13,&H18
20160 REM
20165 REM  ** SIDE 3 **
20166 REM
20170 MEM #,&H2F
20180 MEM #,&H10,&H0C,2,&H0D,&H55,&HD7,&H61,&HC0,4,&H0C
20190 MEM #,2,&H13,&H47,&H71,&HDD,&HC0,7,0,3,5
20200 MEM #,&H61,&HDB,&HF1,&H6E,&HC1,&H0B,&H64,0,&H41,&H53
20210 MEM #,&H0E,&HFB,&H40,&HDE,1,&H60,0,&H5C,3,&H5F
20220 MEM #,&H49,&HA5,&H1F,4,1,&H18
20230 REM
20235 REM  ** SIDE 4 **
20236 REM
20240 MEM #,&H2F
20250 MEM #,0,&H38,&H22,5,&H0D,&H6E,&H13,&HA1,&H13,0
20260 MEM #,&HC0,6,&H1B,&H21,9,&H41,&H45,&H47,&HBB,&H9C
20270 MEM #,&HE0,0,0,1,&HD3,&H76,&H80,&H41,&H19,&H42
20280 MEM #,&H99,&H40,0,0,0,&HB3,&H19,&HC0,&H12,&H73
20290 MEM #,1,&HA1,5,&H12,1,&H14
20300 REM
30000 REM
30010 REM ｷｬﾗｸﾀ ﾊﾟﾀｰﾝ
30020 REM
30030 MEM PATAD,2,2,&HA8,&HA8,&HA8,&HA8:REM BLOCK
```

```
30040 MEM #,2,1,&HA9,&HA9:REM BLUE BLOCK
30050 MEM #,2,2,&HA0,&HB0,&HA1,&HB1:REM RED STONE
30060 MEM #,2,1,&HAE,&HAF,&H20,&H20:REM BREAK STONE 4
30070 MEM #,2,2,&H20,&H20,&HAC,&HAD:REM BREAK STONE 3
30080 MEM #,2,2,&HA6,&HB6,&HA7,&HB7:REM BREAK STONE 2
30090 MEM #,2,2,&HA4,&HB4,&HA5,&HB5:REM BREAK STONE 1
30100 MEM #,2,2,&HA2,&HB2,&HA3,&HB3:REM BULE STONE
30110 MEM #,2,2,&H64,&H74,&H65,&H75:REM ｴﾋﾞｰﾗ LEFT 2
30120 MEM #,2,2,&H62,&H72,&H63,&H73:REM ｴﾋﾞｰﾗ LEFT 1
30130 MEM #,2,2,&H46,&H56,&H47,&H57:REM ｴﾋﾞｰﾗ UP 2
30140 MEM #,2,2,&H48,&H58,&H49,&H59:REM ｴﾋﾞｰﾗ UP 3
30150 MEM #,2,2,&H68,&H78,&H69,&H79:REM ｴﾋﾞｰﾗ RIGHT 2
30160 MEM #,2,2,&H66,&H76,&H67,&H77:REM ｴﾋﾞｰﾗ RIGHT 1
30170 MEM #,2,2,&H6C,&H7C,&H6D,&H7D:REM ｴﾋﾞｰﾗ DOWN 2
30180 MEM #,2,2,&H6E,&H7E,&H6F,&H7F:REM ｴﾋﾞｰﾗ DOWN 3
30190 MEM #,2,2,&H44,&H54,&H45,&H55:REM ｴﾋﾞｰﾗ UP 1
30200 MEM #,2,2,&H6A,&H7A,&H6B,&H7B:REM ｴﾋﾞｰﾗ DOWN 1
30210 MEM #,2,2,&H60,&H70,&H61,&H71:REM ｴﾋﾞｰﾗ ｷﾉｺ 4
30220 MEM #,2,2,&H4E,&H5E,&H4F,&H5F:REM ｴﾋﾞｰﾗ ｷﾉｺ 3
30230 MEM #,2,2,&H4C,&H5C,&H4D,&H5D:REM ｴﾋﾞｰﾗ ｷﾉｺ 2
30240 MEM #,2,2,&H4A,&H5A,&H4B,&H5B:REM ｴﾋﾞｰﾗ ｷﾉｺ 1
30250 MEM #,2,2,&H04,&H14,&H05,&H15:REM ｴﾋﾞｰﾗ SLEEP 1
30260 MEM #,2,2,&H06,&H16,&H07,&H17:REM ｴﾋﾞｰﾗ SLEEP 2
30270 MEM #,2,2,&H08,&H18,&H09,&H19:REM BULE ｴﾋﾞｰﾗ 1
30280 MEM #,2,2,&H0A,&H1A,&H0B,&H1B:REM BULE ｴﾋﾞｰﾗ 2
30290 MEM #,2,2,&H0C,&H1C,&H0D,&H1D:REM BULE ｴﾋﾞｰﾗ 3
30300 MEM #,2,2,&H0E,&H1E,&H0F,&H1F:REM BULE ｴﾋﾞｰﾗ 4
30310 MEM #,2,1,&H80,&H90:REM ｴﾋﾞｰﾗ ﾂﾌﾞﾚ
30320 MEM #,2,2,&H84,&H94,&H85,&H95:REM ﾕﾆｺｰﾝ LEFT 2
30330 MEM #,2,2,&H82,&H92,&H83,&H93:REM ﾕﾆｺｰﾝ LEFT 1
30340 MEM #,2,2,&H88,&H98,&H89,&H99:REM ﾕﾆｺｰﾝ RIGHT 2
30350 MEM #,2,2,&H86,&H96,&H87,&H97:REM ﾕﾆｺｰﾝ RIGHT 1
30360 MEM #,2,2,&H8A,&H9A,&H3E,&H3F:REM ﾕﾆｺｰﾝ SLEEP 1
30370 MEM #,2,2,&H40,&H50,&H41,&H51:REM ﾕﾆｺｰﾝ SLEEP 2
30380 MEM #,2,2,&H42,&H52,&H43,&H53:REM ﾕﾆｺｰﾝ ｷﾉｺ 4
30390 MEM #,2,2,&H8E,&H9E,&H8F,&H9F:REM ﾕﾆｺｰﾝ ｷﾉｺ 3
30400 MEM #,2,2,&H8C,&H9C,&H8D,&H9D:REM ﾕﾆｺｰﾝ ｷﾉｺ 2
30410 MEM #,2,2,&H4A,&H5A,&H8B,&H9B:REM ﾕﾆｺｰﾝ ｷﾉｺ 1
30420 MEM #,2,1,&H81,&H91:REM ﾕﾆｺｰﾝ ﾂﾌﾞﾚ
30430 MEM #,2,2,&HBC,&HBE,&HBD,&HBF:REM ﾌﾗｯﾋﾟｰ LEFT 2
30440 MEM #,2,2,&HB8,&HBA,&HB9,&HBB:REM ﾌﾗｯﾋﾟｰ LEFT 1
30450 MEM #,2,2,&HCC,&HDC,&HCD,&HDD:REM ﾌﾗｯﾋﾟｰ UP 2
30460 MEM #,2,2,&HCE,&HDE,&HCF,&HDF:REM ﾌﾗｯﾋﾟｰ UP 3
30470 MEM #,2,2,&HC2,&HD2,&HC3,&HD3:REM ﾌﾗｯﾋﾟｰ RIGHT 2
30480 MEM #,2,2,&HC0,&HD0,&HC1,&HD1:REM ﾌﾗｯﾋﾟｰ RIGHT 1
30490 MEM #,2,2,&HC6,&HD6,&HC7,&HD7:REM ﾌﾗｯﾋﾟｰ DOWN 2
30500 MEM #,2,2,&HC8,&HD8,&HC9,&HD9:REM ﾌﾗｯﾋﾟｰ DOWN 3
30510 MEM #,2,2,&HE0,&HF0,&HE1,&HF1:REM ﾌﾗｯﾋﾟｰ ｷﾉｺ LEFT 1
30520 MEM #,2,2,&HE2,&HF2,&HE3,&HF3:REM ﾌﾗｯﾋﾟｰ ｷﾉｺ LEFT 2
30530 MEM #,2,2,&HEC,&HFC,&HED,&HFD:REM ﾌﾗｯﾋﾟｰ ｷﾉｺ UP 1
30540 MEM #,2,2,&HEE,&HFE,&HEF,&HFF:REM ﾌﾗｯﾋﾟｰ ｷﾉｺ UP 2
30550 MEM #,2,2,&HE4,&HF4,&HE5,&HF5:REM ﾌﾗｯﾋﾟｰ ｷﾉｺ RIGHT 1
30560 MEM #,2,2,&HE6,&HF6,&HE7,&HF7:REM ﾌﾗｯﾋﾟｰ ｷﾉｺ RIGHT 2
30570 MEM #,2,2,&HE8,&HF8,&HE9,&HF9:REM ﾌﾗｯﾋﾟｰ ｷﾉｺ DOWN 1
30580 MEM #,2,2,&HEA,&HFA,&HEB,&HFB:REM ﾌﾗｯﾋﾟｰ ｷﾉｺ DOWN 2
30590 MEM #,2,2,&HCA,&HDA,&HCB,&HDB:REM ﾌﾗｯﾋﾟｰ UP 1
30600 MEM #,2,2,&HC4,&HD4,&HC5,&HD5:REM ﾌﾗｯﾋﾟｰ DOWN 1
30610 MEM #,2,2,&H02,&H12,&H03,&H13:REM ﾌﾗｯﾋﾟｰ ｺｳｶ
30620 MEM #,2,1,&H00,&H10:REM ﾌﾗｯﾋﾟｰ DIE 1
30630 MEM #,2,1,&H01,&H11:REM ﾌﾗｯﾋﾟｰ DIE 2
30640 MEM #,2,2,&H20,&H20,&H20,&H20:REM NULL PAT.
30650 MEM #,2,1,&H20,&H20:REM NULL PAT.2
30660 MEM #,2,2,&H20,&H20,&H81,&H91:REM ﾕﾆｺｰﾝ ﾂﾌﾞﾚ 2
30670 MEM #,2,2,&H20,&H20,&H80,&H90:REM ｴﾋﾞｰﾗ ﾂﾌﾞﾚ 2
30680 MEM #,1,1,&H21
30690 MEM #,1,1,&H20
30700 MEM #,2,1,&HAA,&HAB
30710 MEM #,2,2,&H20,&H20,&HAA,&HAB
30720 MEM #,2,2,&H20,&H20,&H00,&H10
30730 MEM #,12,1,32,32,32,32,32,32,32,32,32,32,32,32
40000 GOSUB 19000:RETURN
```

dB-IBASIC/dB-COMPILER 使用上の注意

① PC-8801用のIBASIC、COMPILERは、PC-8801用のハードにあわせて作ったものですが、ハードの制約面からdB-IBASICとdB-COMPILERの同居ができず、コンパイル中にエラーが生じた場合は必ずdB-IBASICを再度ロードしてデバックするようになっています。

② また、インタプリタおよびコンパイラで作成したオブジェクトの実行中は割込みが禁止され、64KRAMモードで実行されていますので、ROM内ルーチンを直接コールすることはできません。(どうしてもROM内ルーチンを使用する場合は、マシン語で直接バンク切換えをするようなプログラムを作って行ってください。)

③ ソフトウェアPCGを使用するプログラムは、そのつどソースを読込み実行して定義するのは面倒なので、一度実行後はモニタモードにして7800H番地から8FFFH番地までをセーブしておき、使用時にオブジェクトだけをロードすると便利です。ただし、セーブする際テキストウインドウと、裏RAM（7800H～7FFFH）には注意してください。（セーブする前に必ずモニタモードから］O70、80 ⏎ を実行してください）

④ オブジェクトを一本化する場合は、必ずランタイムパッケージもセーブしなければならないのでかなり長いプログラムになります。ランタイムパッケージは、9800H～BFFFHまでです。 この一本化されたオブジェクトを実行する場合も、スタートアドレスは9803H番地です。なお、一本化されたプログラムを無断で商品化することはできません。

⑤ コンパイラで作成されたオブジェクトは裏RAM上にあるので、そのオブジェクトを見る場合は、テキストウインドウを使ってください。

ex. A＝B＋C＊Dは、次のようなオブジェクトとして作成されます。

```
LD  HL,〔VARC〕
LD  DE,〔VARD〕
CALL  MULT
LD  DE,〔VARB〕
ADD  HL, DE
LD 〔VARA〕, HL
```

注. VARA、VARB、VARC、VARDは変数のアドレス。
MULTはランタイムパッケージ内のかけ算ルーチン。

その他、定数の四則演算にも高速化をはかっています。

⑥ コンパイラはできるだけ最適化をしていますが、特に文字列の２次元配列を使用すると効率が悪くなります。できるだけ文字列の大きさは 2^n-1 の値にして（＄Lで設定）１次元で行うようにしてください。

評価用ミニフラッピーの面パターンは４画面です。
ゲームのあそび方●２４６８キーにより、フラッピーが上下左右に移動します●スペースキーできのこを敵にぶつけて眠らせます●岩を押し進めて道をつくったり、落下させて敵をつぶします。ただし、反対側に何かあると壊れてしまいます●フラッピーは右利きです。

「PC-8801用200画面フラッピー」dB-SOFTより発売中！／4,500円

PC-8801mkII/PC-8801

dB-IBASICコンパイラ・ブック
発行日　昭和59年2月1日　第1版　第1刷
発行人　古谷貞行
企画制作
著　　作　デービーソフト技術部
発行所　デービーソフト株式会社
〒060 札幌市中央区北3条西2丁目
カミヤマビル☎(011)251-7462

編　集／デービーソフト 企画室

御注意

①本書又はプログラムの内容の一部又は全部を無断で転載・コピーすることは著作権上固くお断わりします。
②本書又はプログラムの内容に関して、将来予告なしで変更することがあります。
③本書の内容については万全を期して作成いたしましたが、万一御不審な点や誤り、記載もれなど、お気付きの点がありましたら御一報下さい。
④本プログラムについては万全を期しておりますが、万一当社の製造上の原因による不良がありましたら新しい製品とお取り替えいたします。
⑤運用した結果の影響につきましては、責任を負いかねますので御了承願います。

dB-SOFT

8801

dB-1 BASIC & dB-COMPILER SERIES

企画制作
著　　作 **デービーソフト**
〒060 札幌市中央区北3条西2丁目カミヤマビル ☎(011)251-7462

N3-L0022

COPYRIGHT ©1984 dB-SOFT INC.　NO PART OF THIS PROGRAM SHOULD BE USED WITHOUT PERMISSION